

Tutorial: Creating A FoxPro Application

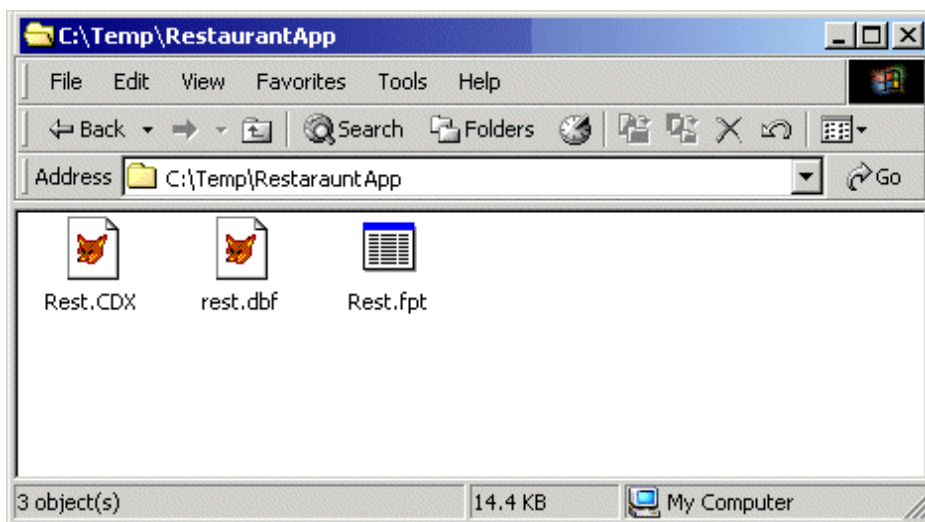
Dr. Thomas E. Hicks
Computer Science Department
Trinity University

Organize Files & Create Directories/Folders

Download and decompress the following FoxPro Database Files into directory C:\Temp\RestaurantApp

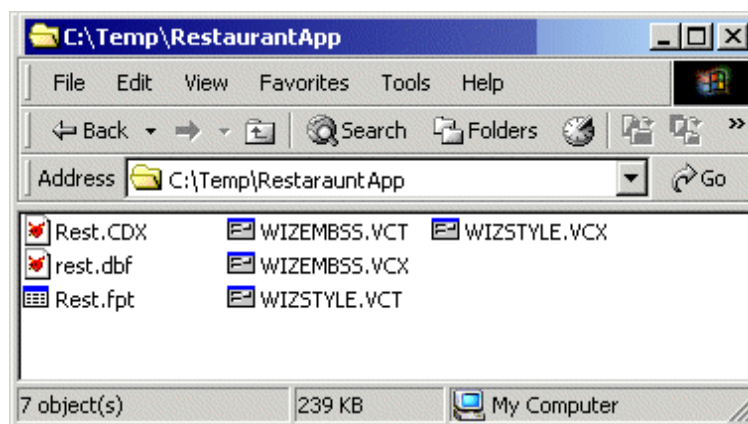
[Restaurant Application. zip](#)

1] The contents of the initial folder should be the Rest.dbf, Rest.cdx, and Rest.fpt.(See Below!)



2] In order to easily find the wizards that we are going to use on any computer, I recommend placing at least the four screen wizard file in your directory.

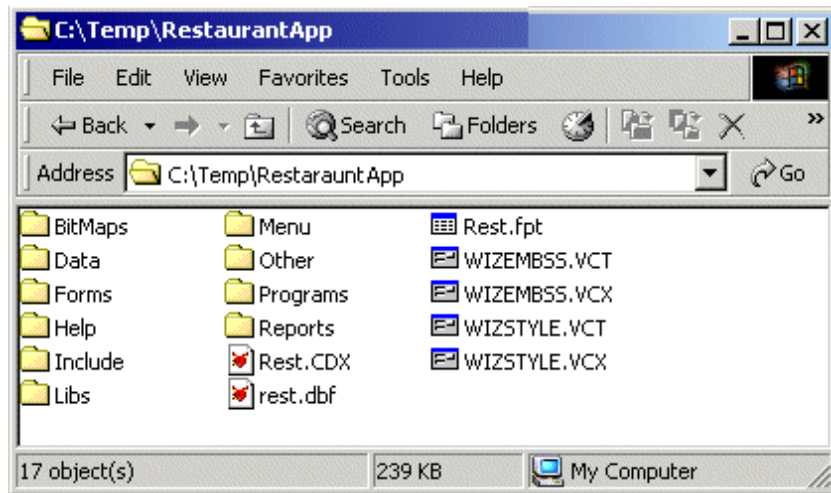
[Wizards.zip](#)



3] In order to keep from having dozens of files at the root level, it is best to organize the various files for an application. Some of them will have to be kept in the root directory, but others we shall organize by topic. For example, we shall place all of our images in the BitMaps directory. We shall place all of our forms in the Forms directory. We shall place all of our Reports in the Reports directory. We shall place all of our database tables and indexes in the Data directory. We shall place all of our program code in the Programs directory.

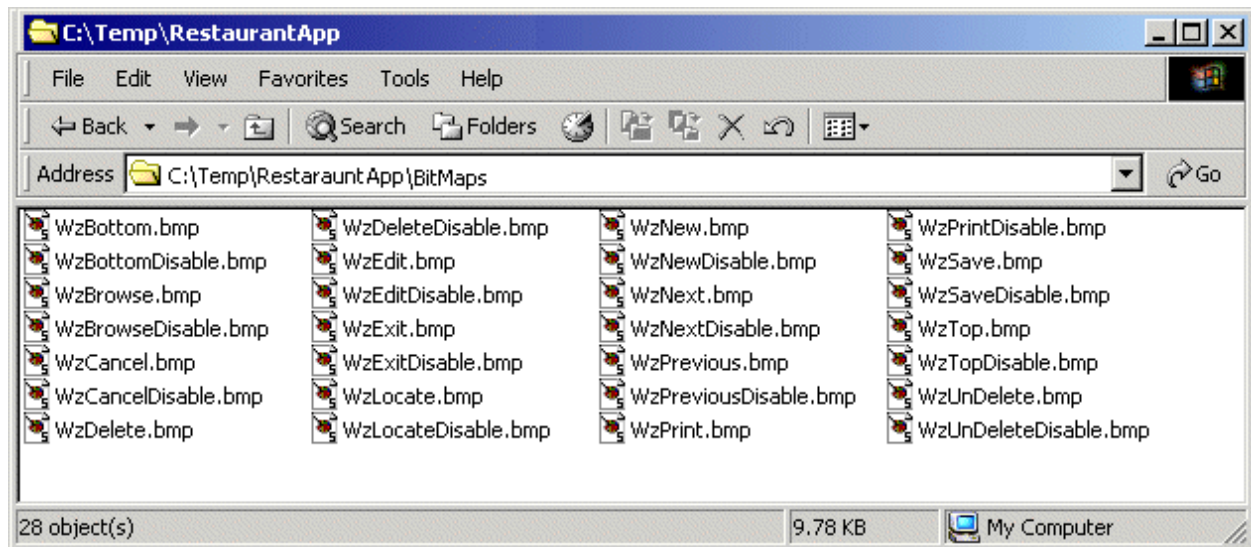
Create all of the directories that you see below. This is a very simple database application with no relationships and only a single file. For an application as simple at this, we shall not yet use the Help, Include, Libs, Menu, or Other directories. See

Below.

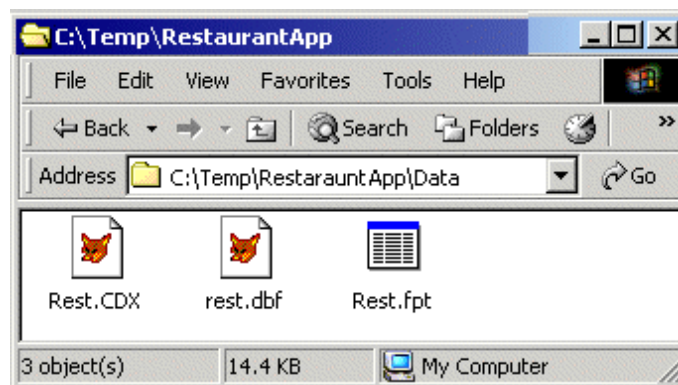


4] I am including several of the bitmap images that I use for applications in the file below. Extract them and add them to the BitMaps folder. (See Below!)

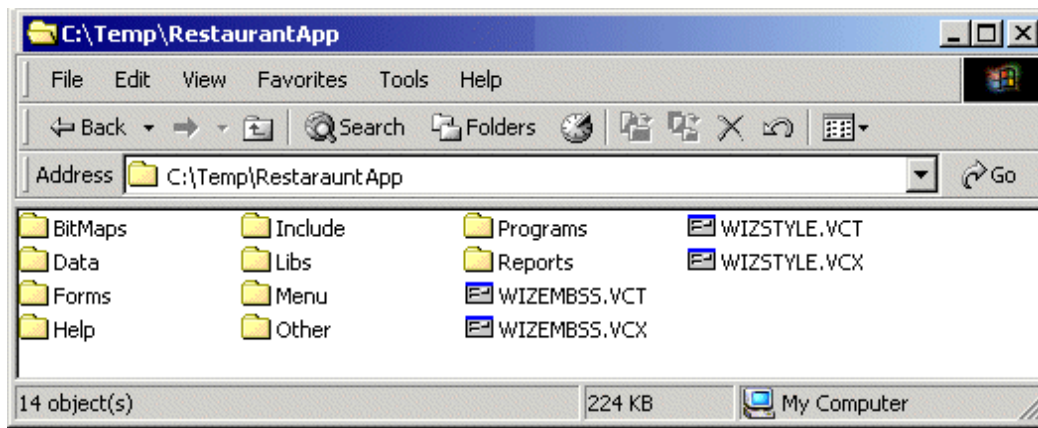
[BitMaps.zip](#)



5] Move the three rest files into the Data folder. (See Below!)

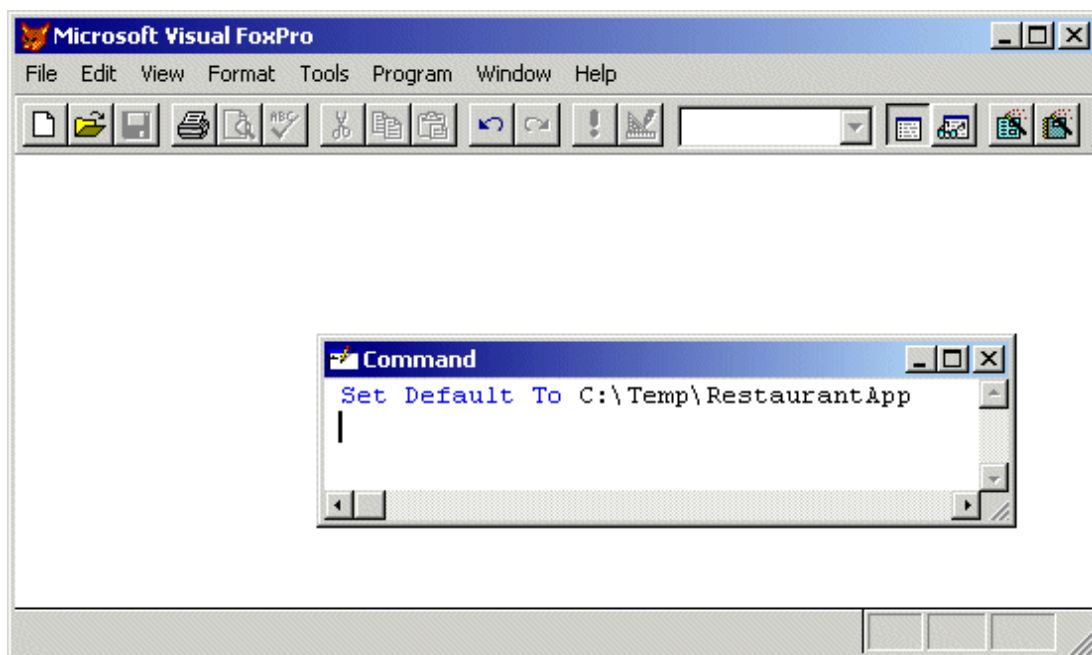


6] This should leave the following files/directories at the root level. (See Below!)

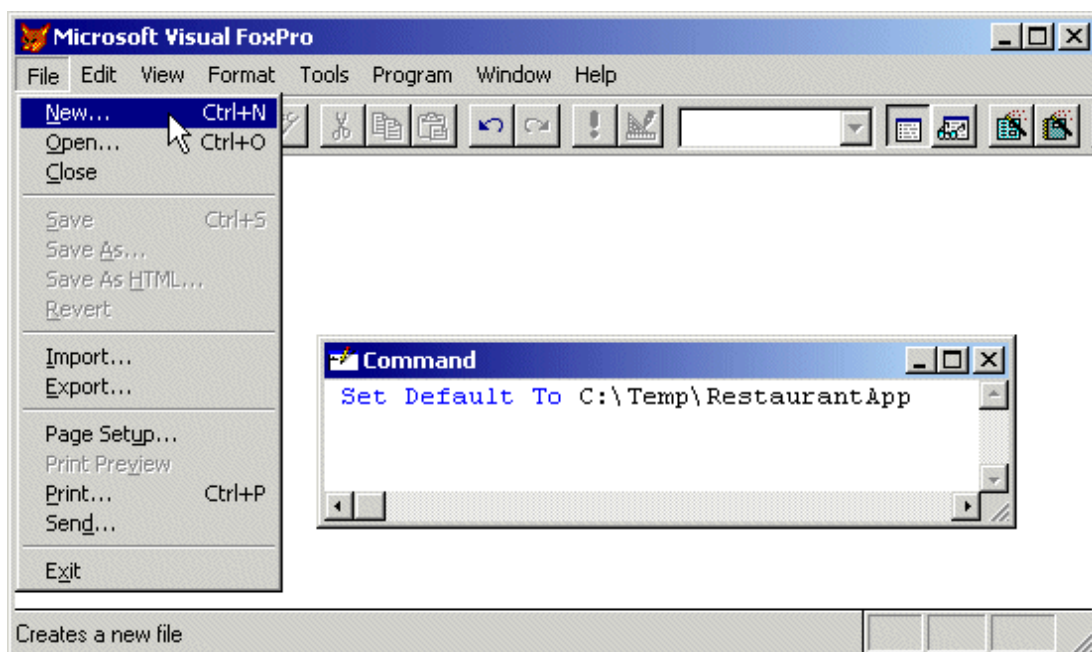


Create A Database Project

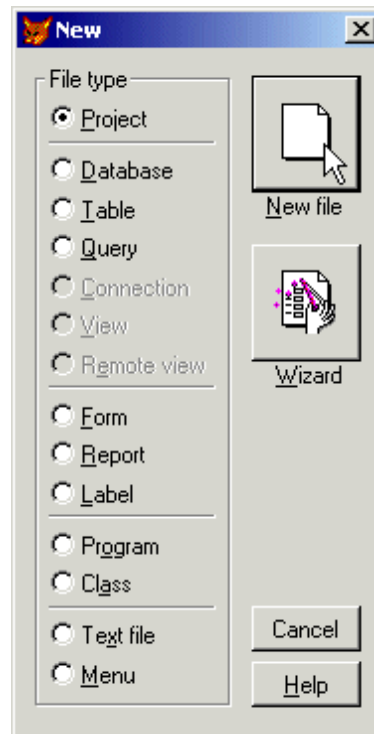
1] Start FoxPro. Important==> Set Default to C:\Temp\RestaurantApp! (See Below!)



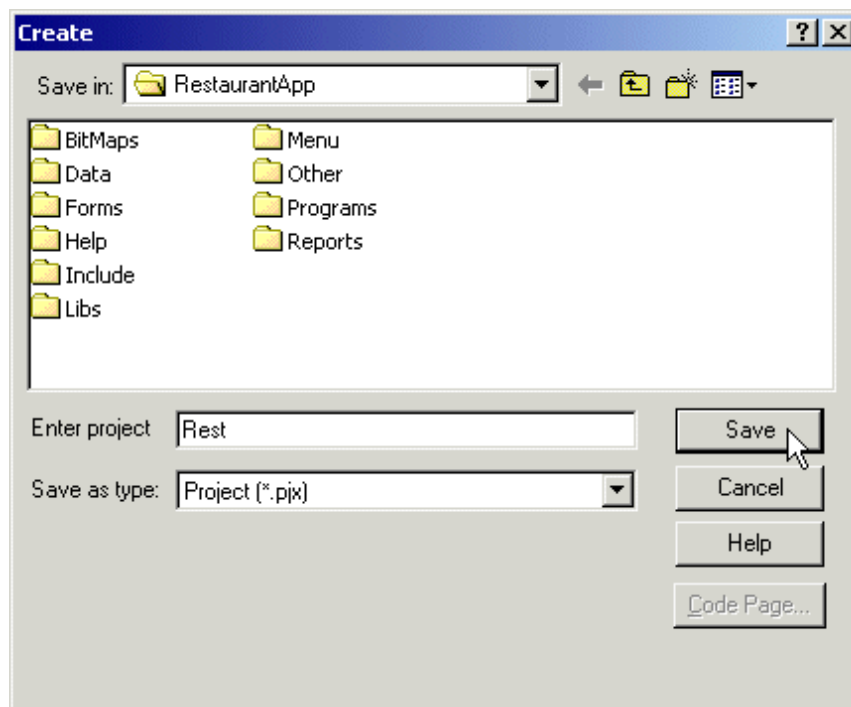
2] Using the mouse, hold down the File Menu and select New (See Below!)



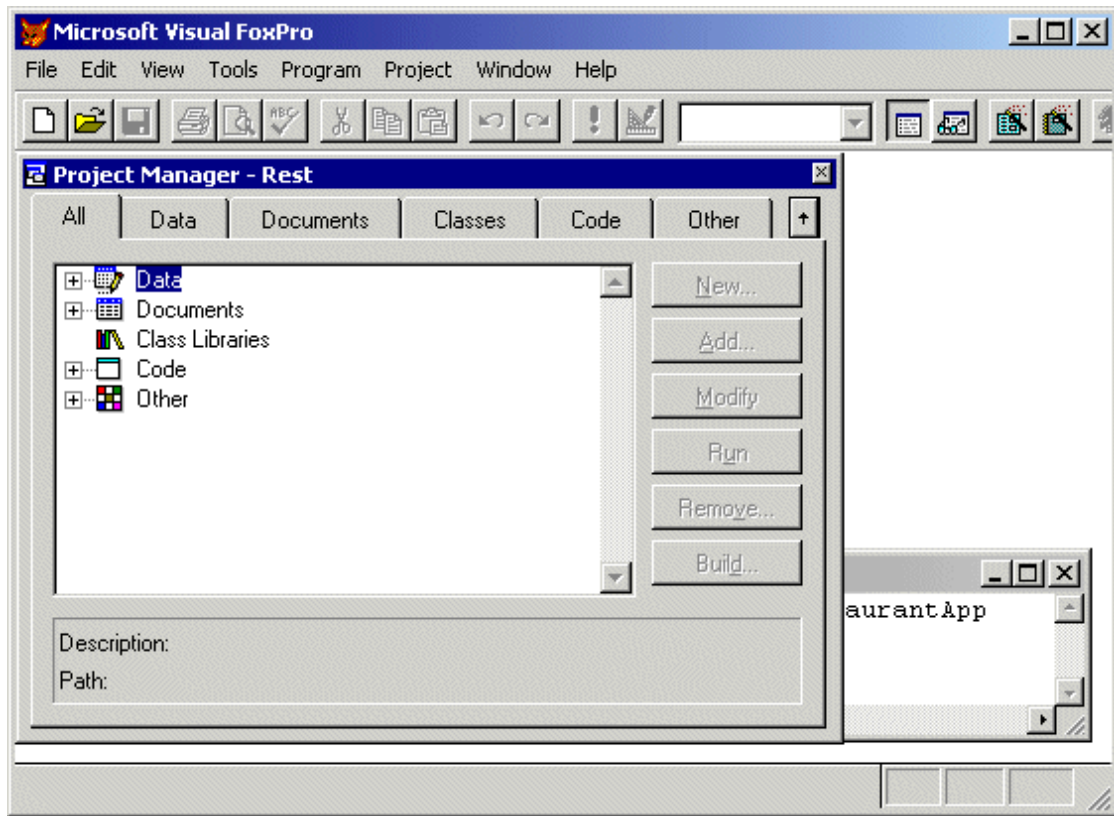
2] Using the mouse, select the Project Radio button. Using the mouse, push the New File button. (See Below!)



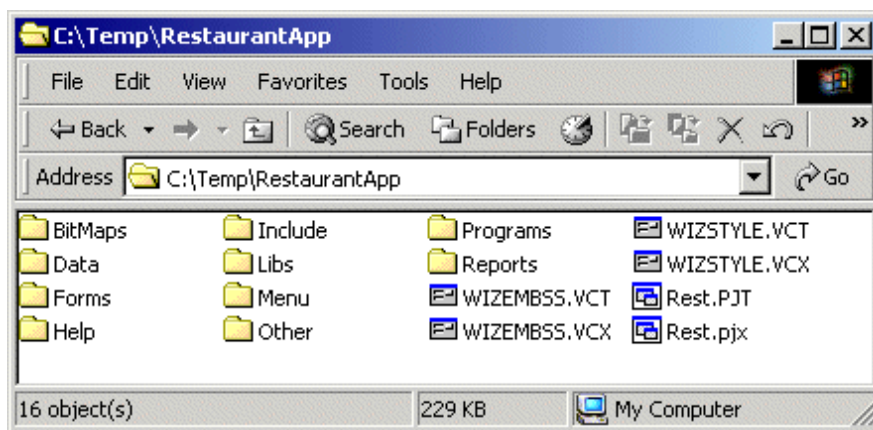
3] If you have set the default properly, the RestaurantApp folder should be in the Save In bar at the top; Enter a project name of Rest. Using the mouse, select/push the Save button.



4] The project is now created. (See Below!)

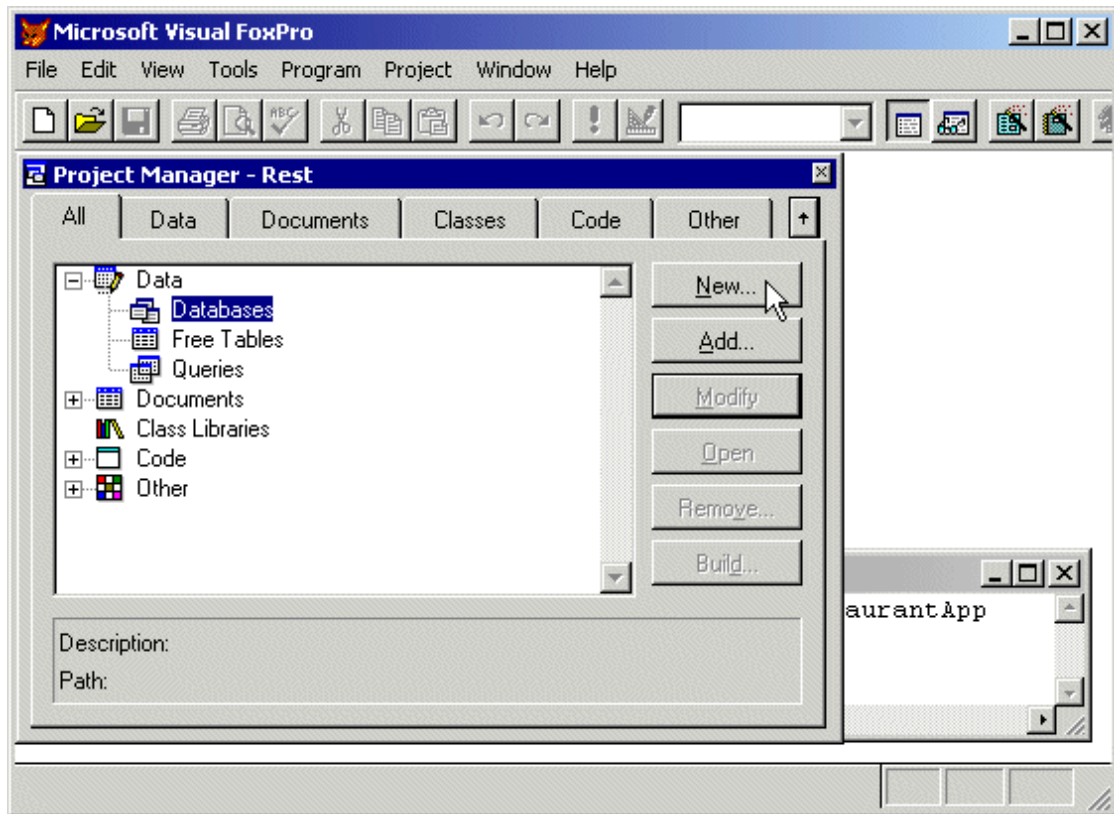


5] This new project should have added files Rest.PJT and Rest.pjx to the root of the RestaurantApp folder. Check it out! (See Below!)

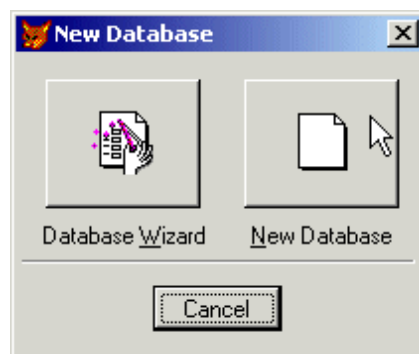


Create A Database

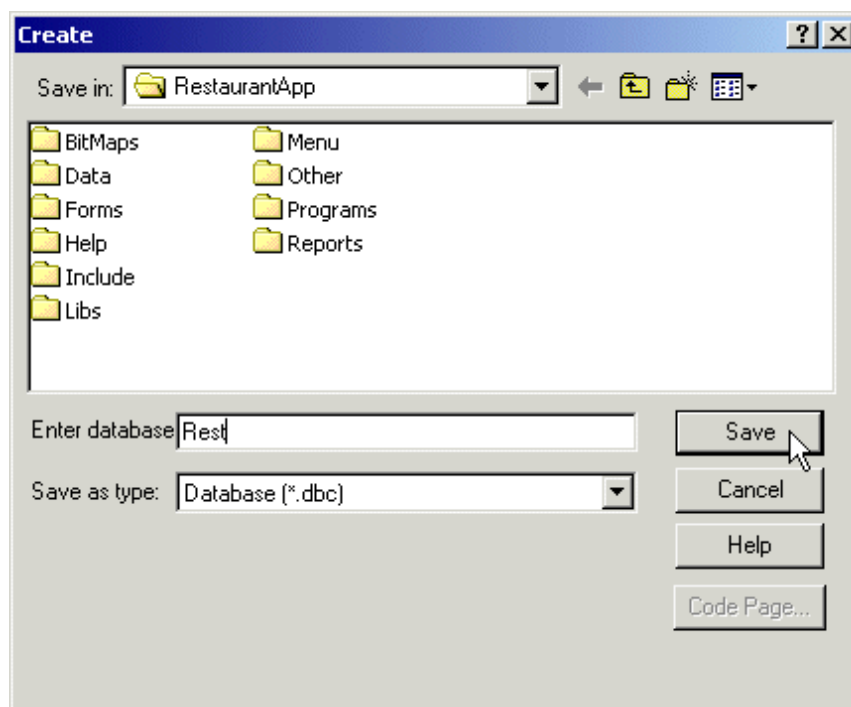
1] Expand Data or select the data tab. Select Database. Using the mouse, select/push the New button. (See Below!)



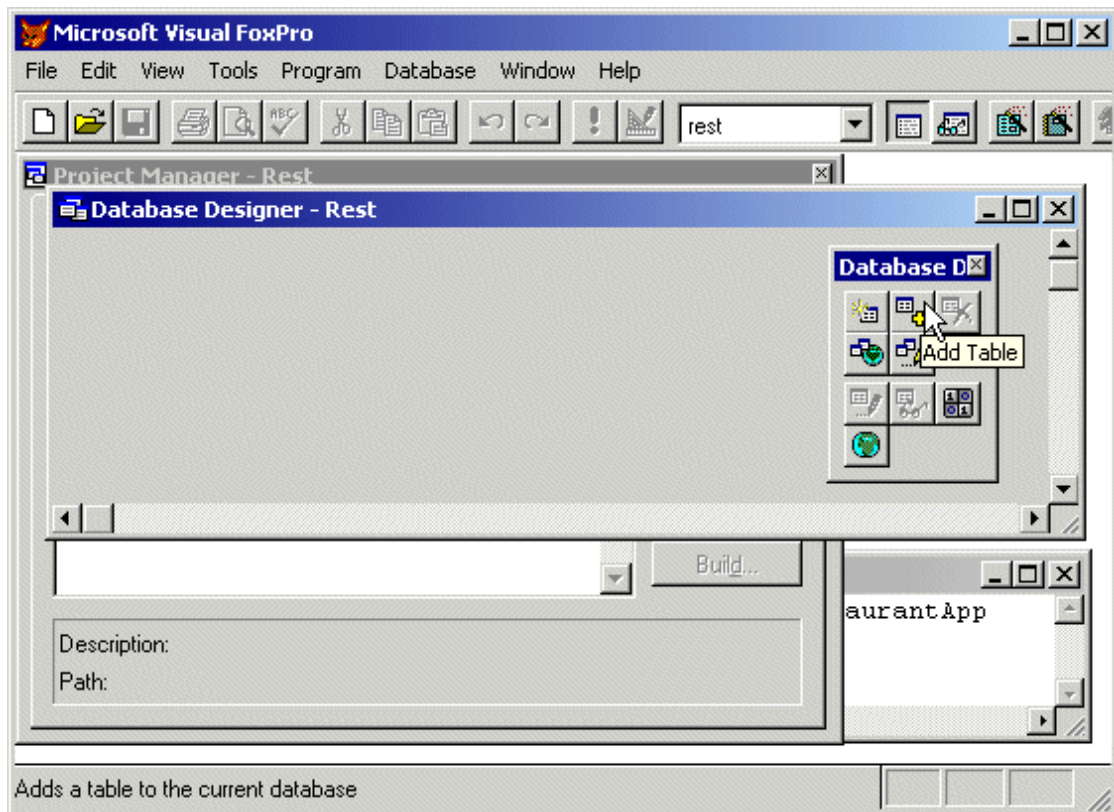
2] Using the mouse, select/push the New Database button. (See Below!)



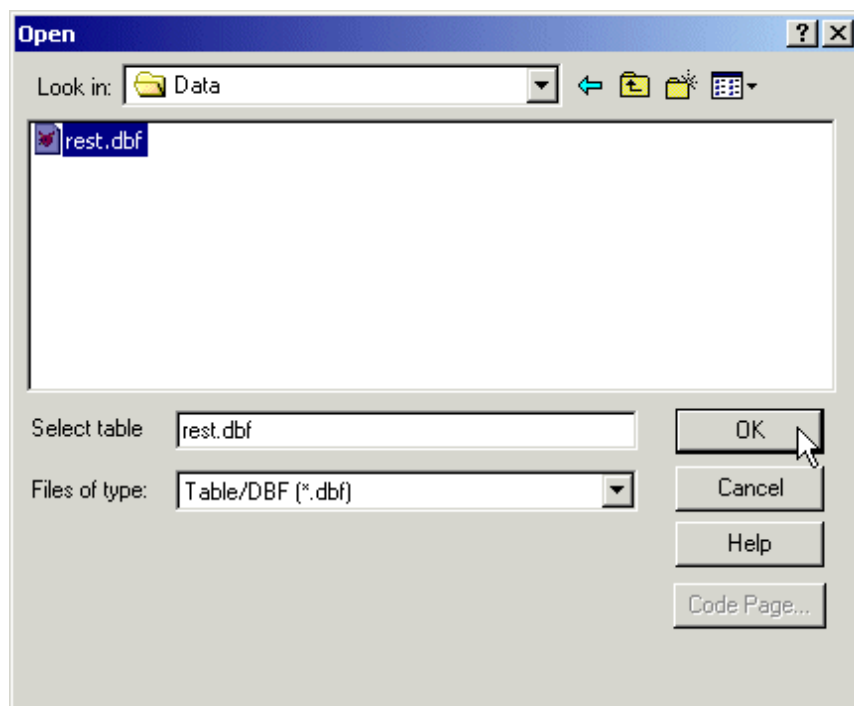
3] Enter Rest in the Enter Database dialog box. Using the mouse, select/push the Save button. (See Below!)



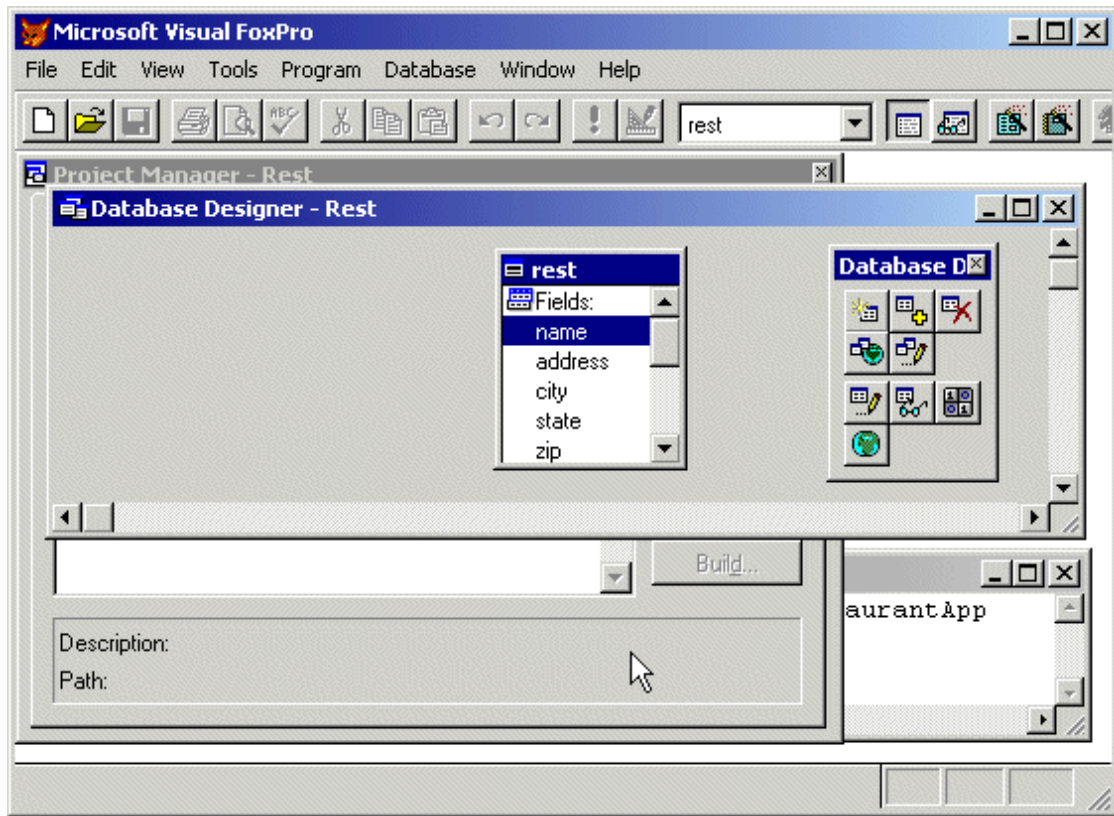
4] We now want to add all of our database tables to the database. Using the mouse, select/push the Add Table button.(See Below!)



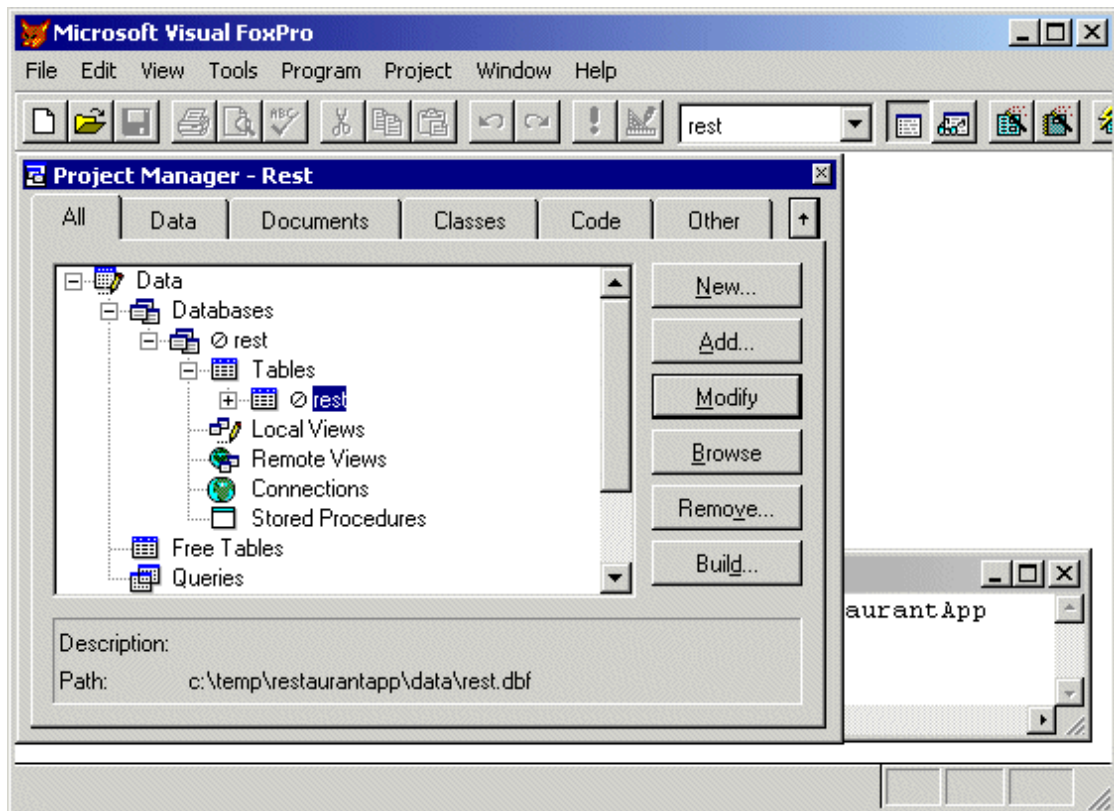
5] Go to the Data folder and select each of your tables; only one in this application. (See Below!)



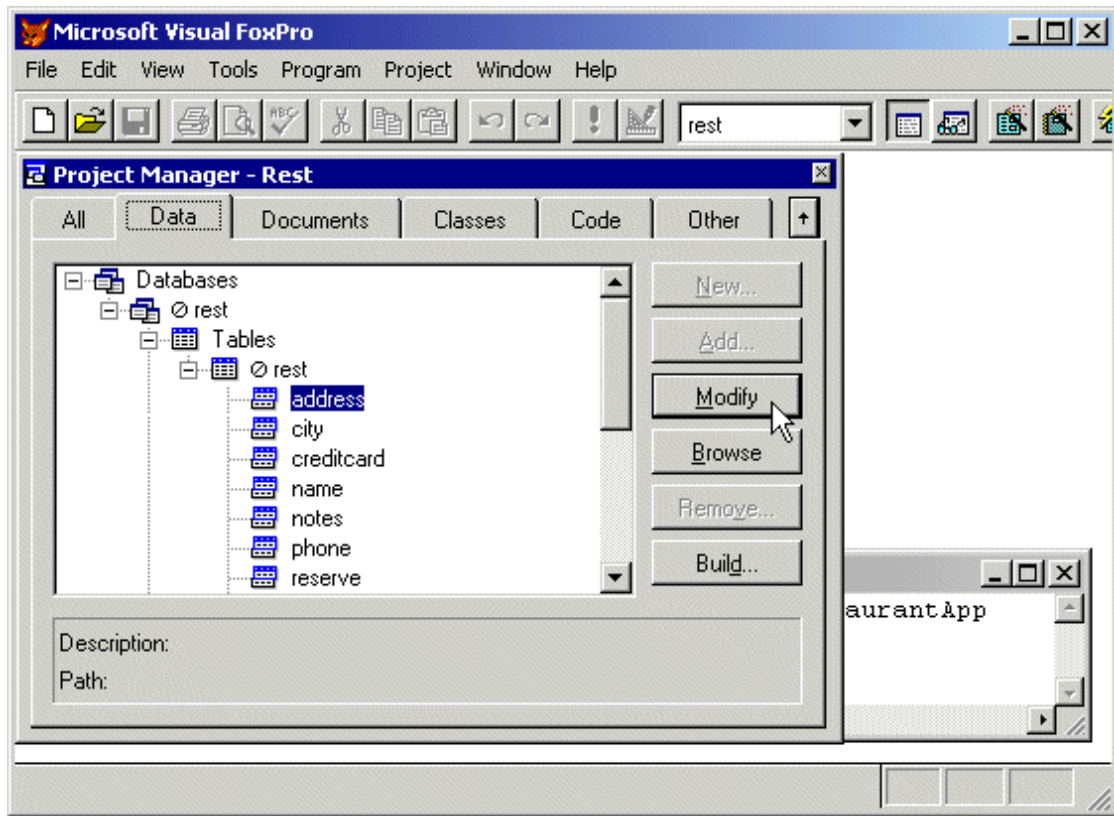
6] You database now has the Rest Table associated with it! (See Below!) You may also create new tables with the New Table button. Tables inside the database are permitted to have much longer field names. Try it!



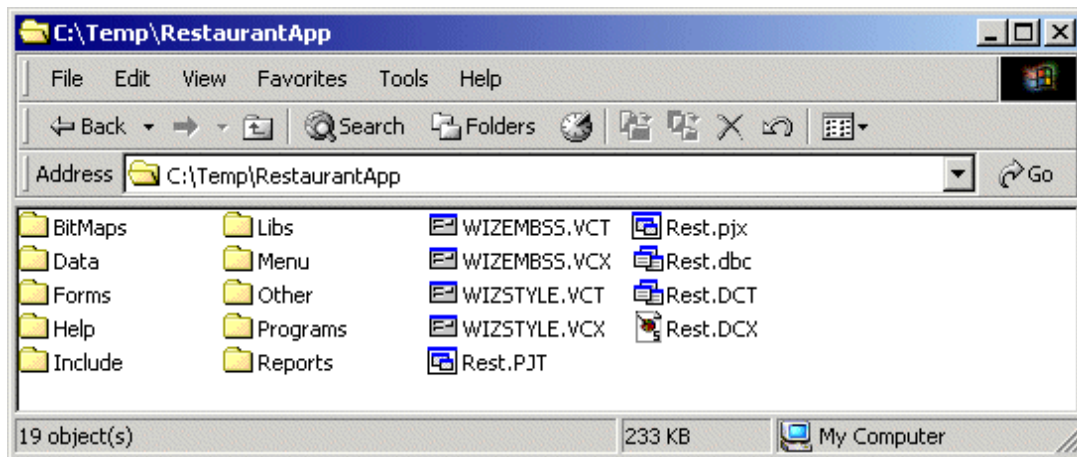
7] The application provides easy point and click access to all databases, all tables, and all fields in the tables.



8] The view below shows the Rest table expanded in the Data tab view. Note that you can modify the table structure or browse the table from this view. (See Below!)

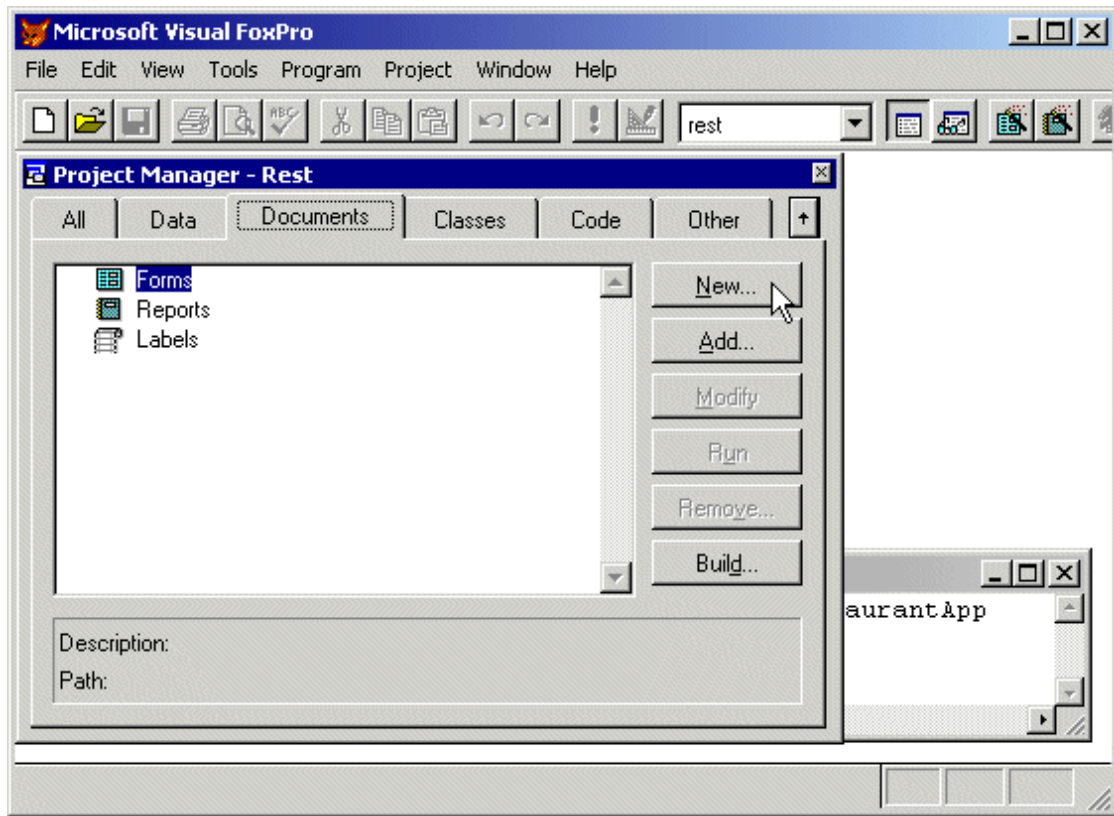


9] Note that this has added files Rest.DCT and Rest.DCX to the root of your application folder. (See Below!)

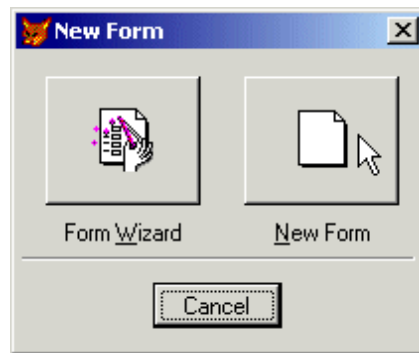


Create The Rest Form

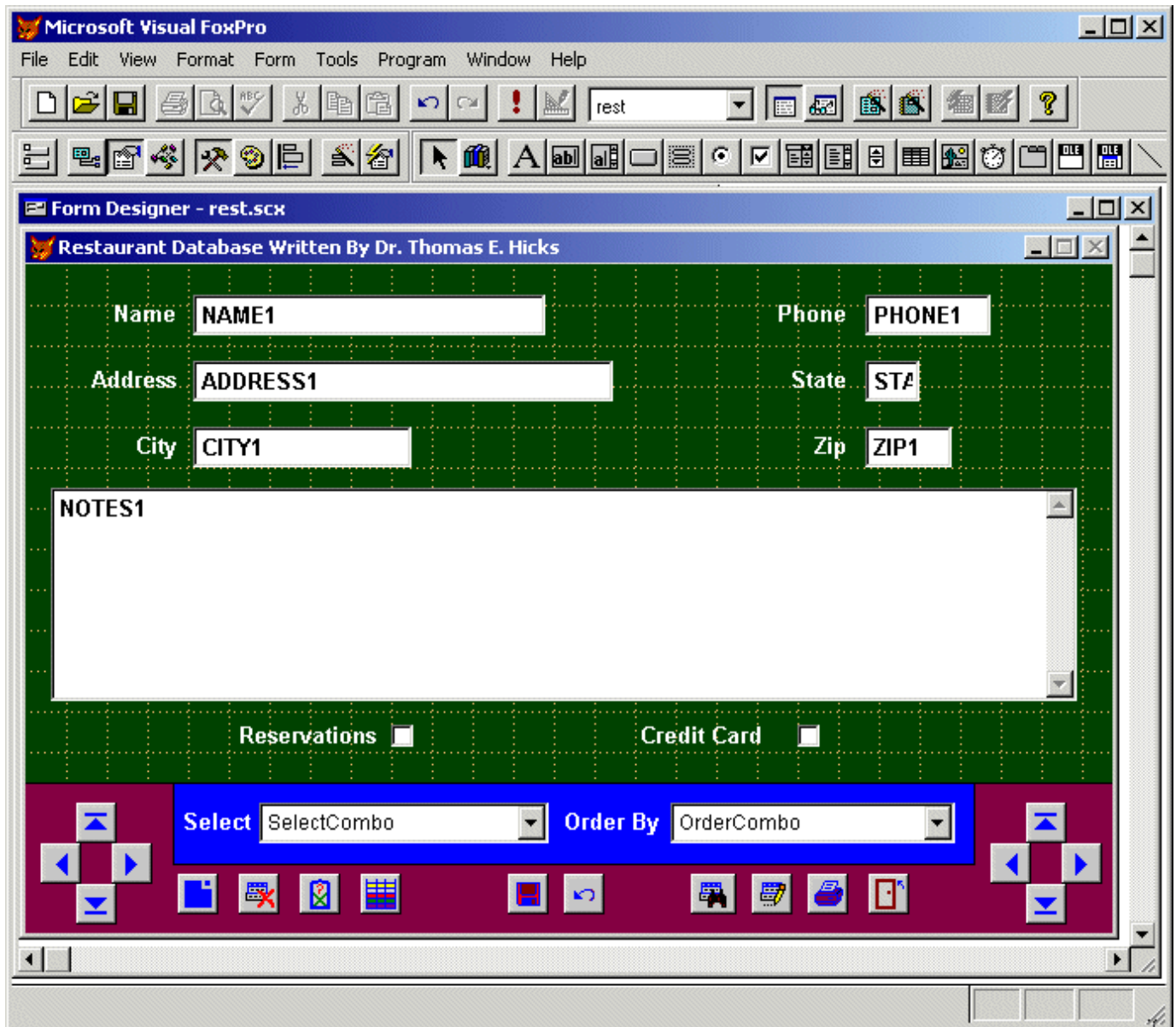
1] Select the Documents Tab. All Reports, Labels, and Forms are to be organized here. Using the mouse, select/push the New button. (See Below!)



2] Using the mouse, select/push the New Form button. (See Below!)



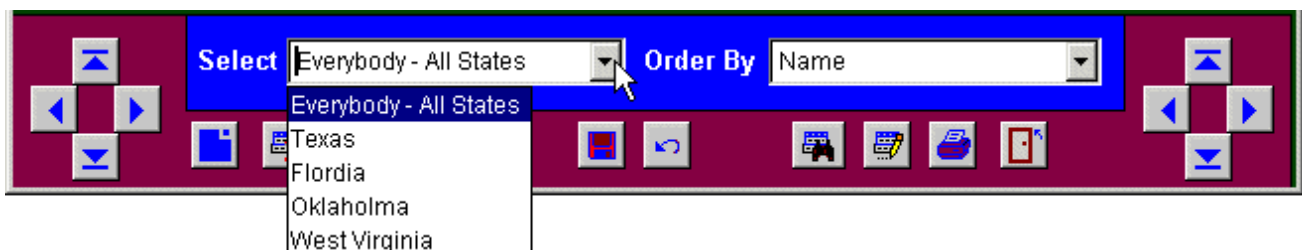
3] It is not the purpose of this tutorial to teach form or report design. Check one of my other tutorials. Create the following form. **Put your name at the menu bar in place of mine.** Save it as Rest in the Forms folder/directory. We shall not use all of the buttons yet! (See Below!)



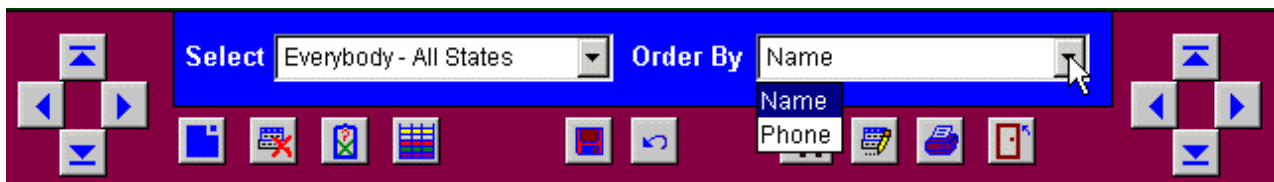
Special Rest Form Settings and Events

- A. Destroy Event ==> application.visible = .T.
- B. Get Focus Event ==> _screen.activeform.refresh
- C. Set Closable to false
- D. Set MaxButton to false
- E. Set MinButton to true
- F. Set ShowWindow to 2 - As Top Level Form
- G. Set WindowState to 0 - Normal
- H. Show Tips = True
- I. Set Scroll Bars = 0 None
- J. AutoCenter = .T.
- K. Check Tab Order. Create Caption! Review the form design guidelines.

4] Your SelectCombo shall have the following 5 options:



5] Your OrderCombo shall have the following 2 options:



Create The Search Form

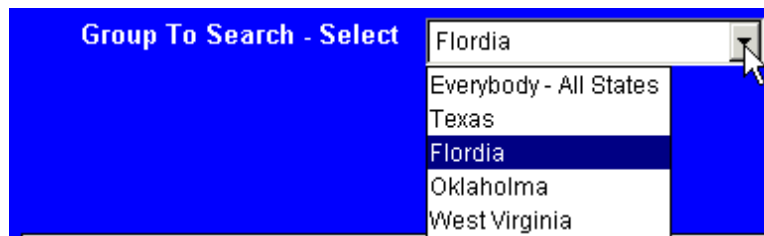
1] Create the following generic Search form with your project manager. Note that it has the same two combos as were on your Rest form. Think copy and paste! The form title bar is to be **Search Engine**.

Special Search Form Settings and Events

- A. Always On Top = .T.
- B. AutoCenter = .T.
- C. Caption = Search Engine
- D. Closable = .F.
- E. GotFocus Event = Do SetSearchExample
- F. MaxButton = .F.
- G. MinButton = .F.
- H. Name = Search
- I. Show Tips = True
- J. Set ShowWindow to 2 - As Top Level Form

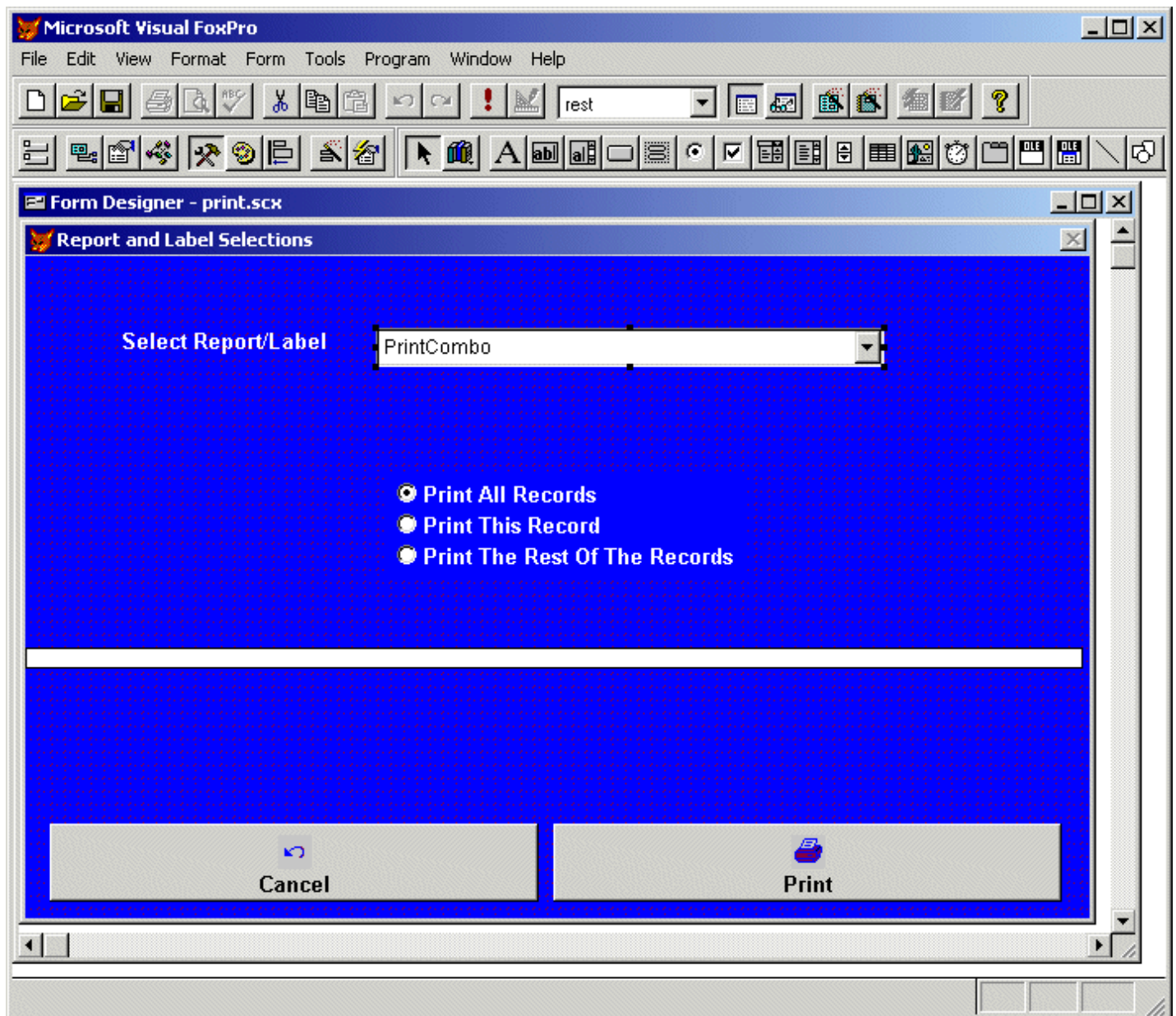
2] Your Search By choices are to have the following 2 options:

3] Your Group To Search - Select choices are to have the following 5 options:



Create The Print Form

1] Create the following generic Print form with your project manager. The combo is to have list of all the reports for your application. The form title bar is to be **Report and Label Selections**. The three required options for the PrintCombo are Full Report [Professionally laid out report containing all of the information about the restaurant is to go to the printer!] and NamePhone Report [Professionally laid out report containing only the Name and Phone of all restaurants is to go to the printer] and a mailing label for envelopes.



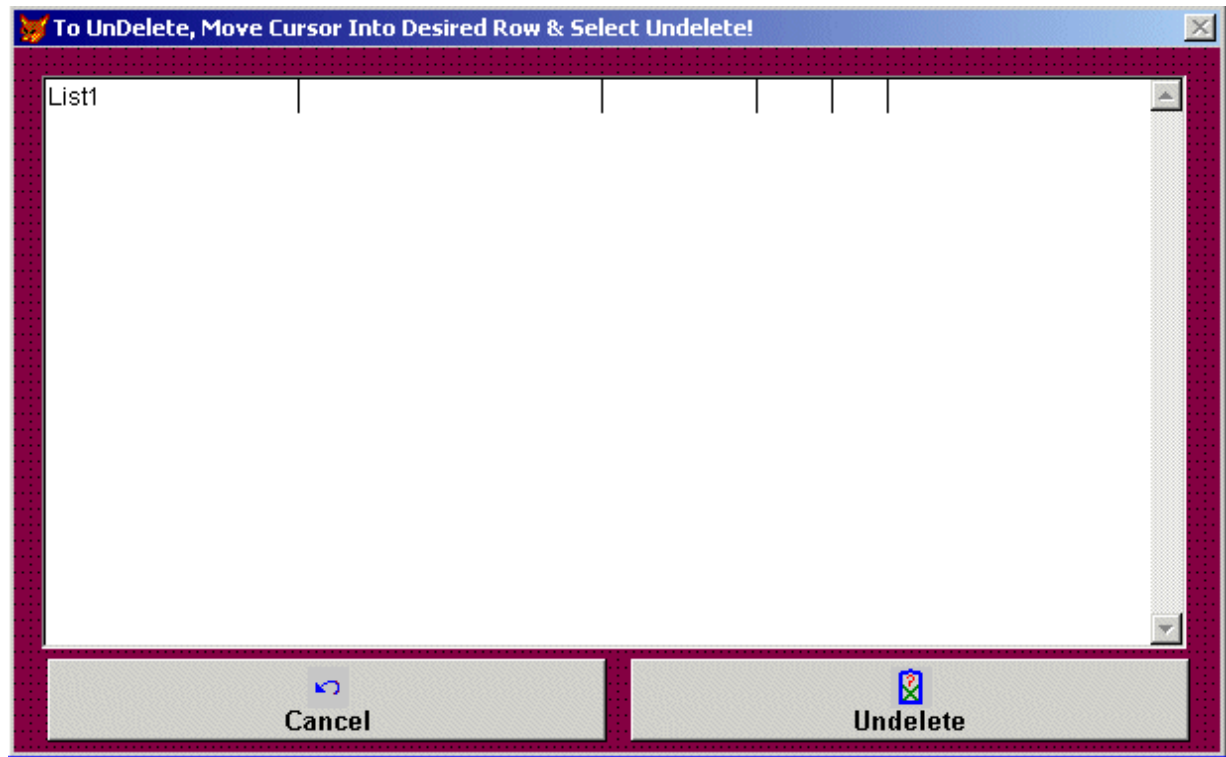
Special Print Form Settings and Events

- A. Always On Top = .T.
- B. AutoCenter = .T.
- C. Caption = Print Report and Label Selections
- D. Closable = .F.
- E. MaxButton = .F.
- F. MinButton = .F.
- G. Name = Print

- H. Show Tips = True
- I. Set ShowWindow to 2 - As Top Level Form

Create The Undelete Form

1] Create the following Undelete form with your project manager. Shown in the List box are to be only those restaurants that are deleted. The form title bar is to be **To Undelete, Move Cursor Into Desired Row & Select Undelete!**



Special Undelete Form Settings and Events

- A. Always On Top = .T.
- B. AutoCenter = .T.
- C. Caption = To UnDelete, Move Cursor Into Desired Row & Select Undelete!
- D. Closable = .F.
- E. MaxButton = .F.
- F. MinButton = .F.
- G. Name = Undelete
- H. Show Tips = True
- I. Set ShowWindow to 2 - As Top Level Form

Reports

1] Create a full report that professionally displays all of the information of the filtered restaurants with page numbers, dates, and captions. Put a divider line between each record. Call the report FullReport. Store the Reports in the Report folder.

2] Create a Name & Phone report that professionally displays the name and phone of the filtered restaurants with page numbers & dates. You may put the captions at the top of the columns if you like. Call the report NamePhone.

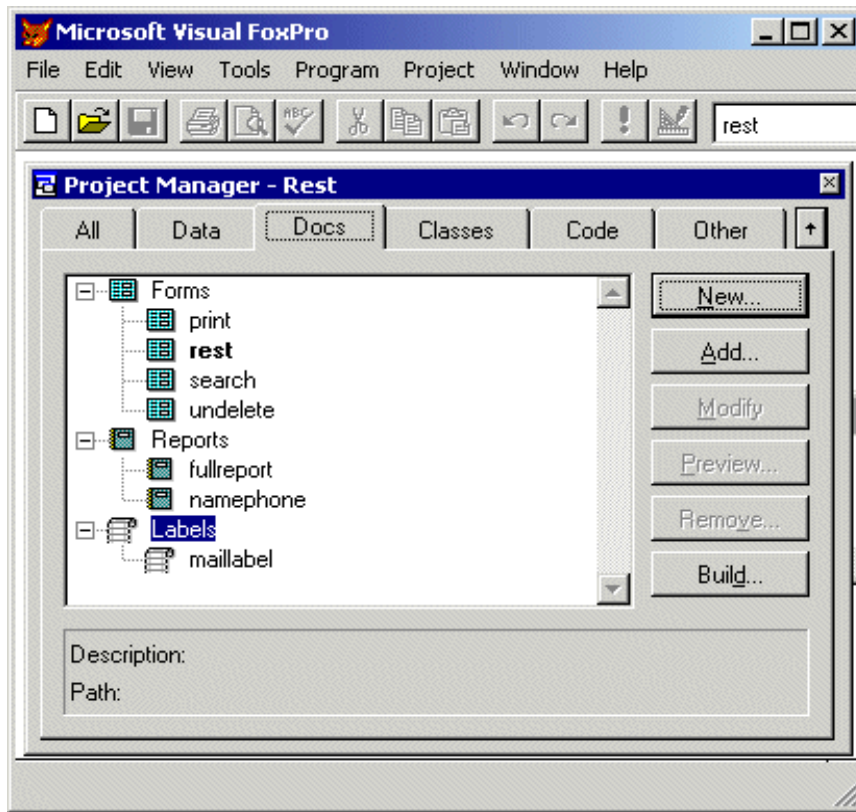
Extra credit <1> pt If you get the filter information on the report title as well.

SKIP Not Installed On Lab - Labels - SKIP Not Installed On Lab

1] Create a mailing label that professionally displays the Name on line 1 of a label. Display the Address on line 2. Display the City, State, and Zip on line 3 of the label. Call the label Mail.

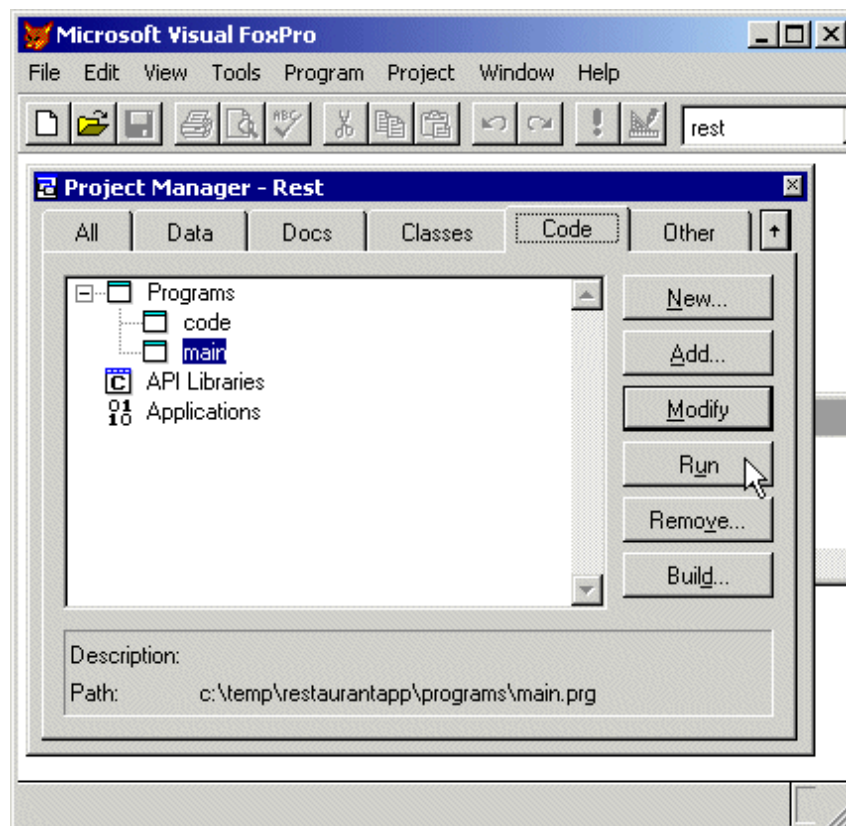
Beneath The Docs Tab

1] Your application should now have the following forms, reports, and labels. (See Below!)

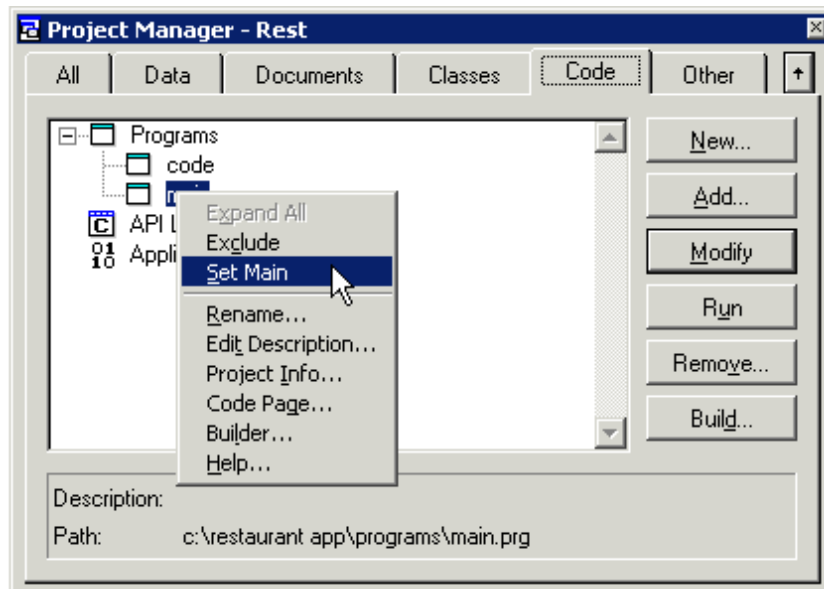


Add Programs To Project

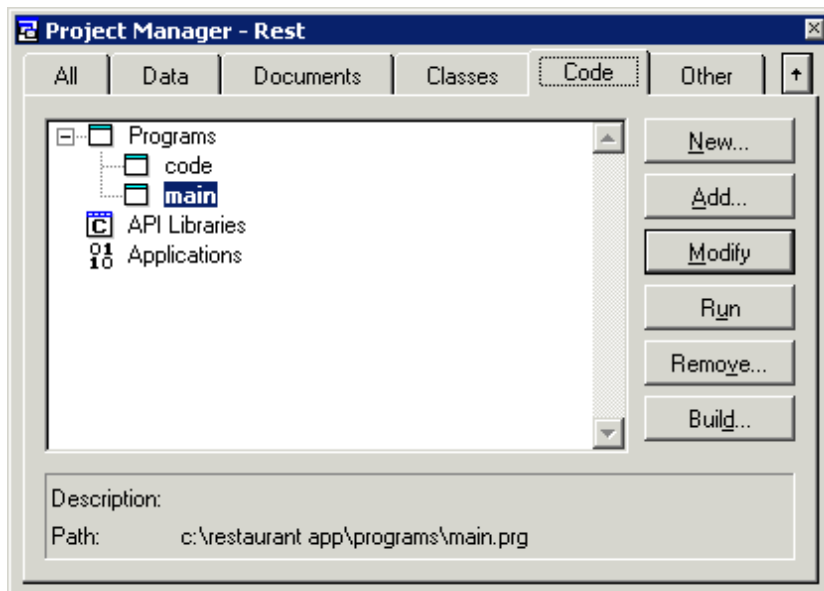
1] Add a main program in which to launch your application and a code file in which to store all of your button code. All forms and applications should be tested by selecting the Main and pushing the Run button. See below!



3] Using the mouse, right click on main and select Set Main. You may also hold down the project menu and select Set Main. (See Below!)

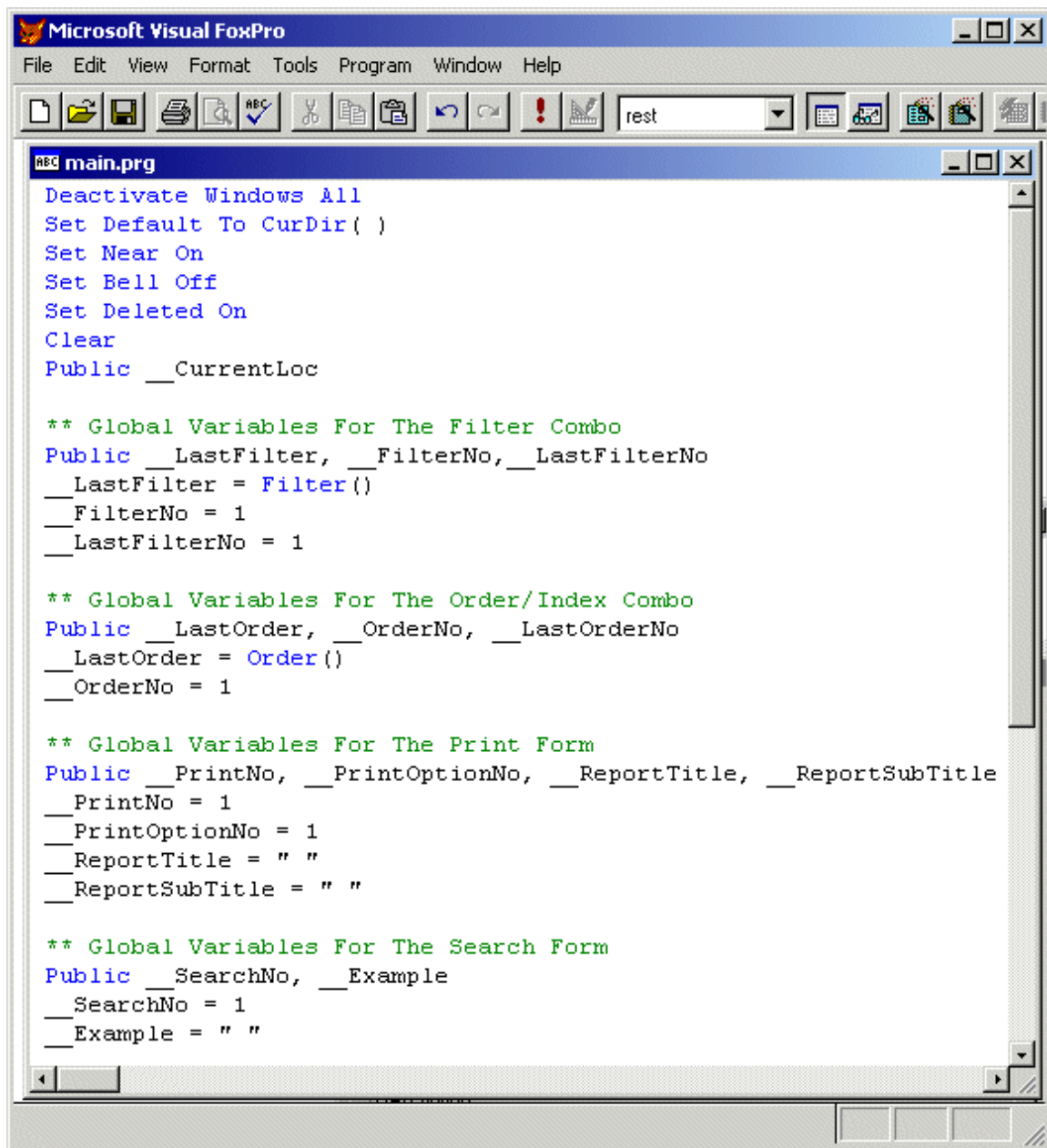


4] Note that it is now bold. This shall be the program that is run when the executable is built. You must do this in order to create executables. (See Below!)

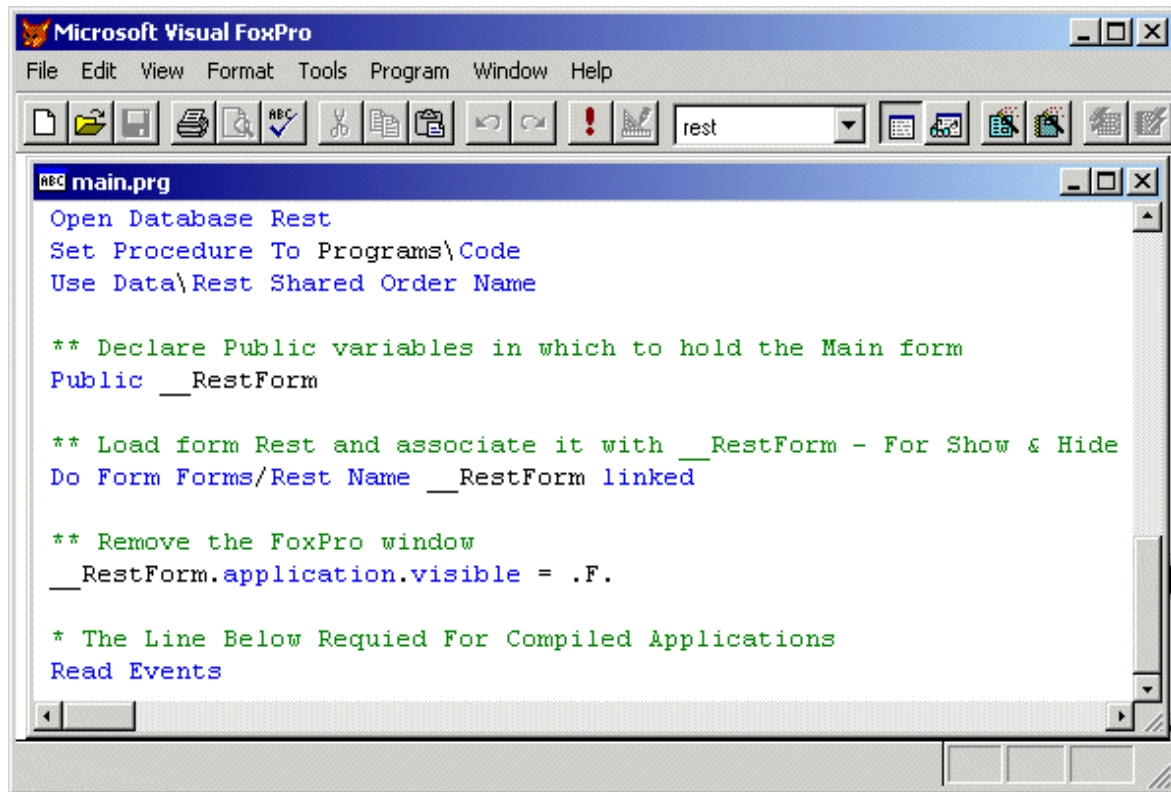


Create A Good Start Program - Main.prg

1] Some of the system variables begin with an underscore(_). By convention, I start all of my global variables, declared in main, with a double underscore(__). It makes them easy to identify. Listed below is a screen capture of one good student's attempt to solve this problem. Depending upon how you implement your solutions, you may or may not need to use all of the same variables. There is nothing new for my students in this block of code.



2] Note that you have to path to the code.prg file that contains all of the button code. You will also have to path to the database table and the forms. (See Below!)



Create Code.PRG To Contain All Button Code

1] The Code to associate with button Next is found below:

```

*-----*
*-                               Procedure Next                               -*
*- Purpose : Load the next logical record in the form. Wrap around -*
*-           from the bottom to the top.                               -*
*- Called From : Valid Event of Next Button.                          -*
*-                               -*
*- Written By: Dr. Thomas E. Hicks                                     App: Visual FoxPro 5 -*
*-----*
Procedure Next
    If EOF() Then
        GoTo Top
    EndIf
    Skip 1
    If EOF() Then
        GoTo Top
    EndIf
    _screen.activeform.refresh
Return
EndProc
  
```

2] The Code to associate with button Previous is found below:


```

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*-                               Procedure Previous                    -*
*-                               -*
*- Purpose : Load the previous logical record in the form. Wrap    -*
*-           around from the top to the bottom.                      -*
*- Called From : Valid Event of Previous Button.                    -*
*-                               -*
*- Written By: Dr. Thomas E. Hicks                                App: Visual FoxPro 5 -*
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
Procedure Previous
    If EOF() Then
        GoTo Top
    EndIf
    Skip -1
    If BOF() Then
        GoTo Bottom
    EndIf
    _screen.activeform.refresh
Return
EndProc

```

T

3] The Code to associate with button Top is found below:

```

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*-                               Procedure Top                        -*
*-                               -*
*- Purpose : Load the first/top logical record in the form.        -*
*- Called From : Valid Event of Top Button.                          -*
*-                               -*
*- Written By: Dr. Thomas E. Hicks                                App: Visual FoxPro 5 -*
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
Procedure Top
    GoTo Top
    _screen.activeform.refresh
Return
EndProc

```

4] The Code to associate with button Bottom is found below:

```

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*-                               Procedure Bottom                    -*
*-                               -*
*- Purpose : Load the last/bottom logical record in the form.      -*
*- Called From : Valid Event of Bottom Button.                      -*
*-                               -*
*- Written By: Dr. Thomas E. Hicks                                App: Visual FoxPro 5 -*
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
Procedure Bottom
    GoTo Bottom
    _screen.activeform.refresh
Return
EndProc

```

5] The Code to associate with button Add is found below:

```
*- *-*-*-* Procedure Add *-*
```

```
*-                                     *-
```

```
*- Purpose : Load a blank form and allow the user to Save or *-
```

```
*-           Cancel. Save will append a new record with the   *-
```

```
*-           appropriate data.                                 *-
```

```
*- Called From : Valid Event of Add Button.                    *-
```

```
*-                                                         *-
```

```
*- Written By: Dr. Thomas E. Hicks                               App: Visual FoxPro 5 *-
```

```
*- *-*-*-* *-*
```

```
Procedure Add
```

```
    Append Blank
```

```
    _screen.activeform.Name1.SetFocus
```

```
    _screen.activeform.refresh
```

```
Return
```

```
EndProc
```

6] The Code to associate with button Delete is found below:

[illegible]

7] The Code to associate with button Exit is found below:

```
*-----*
```

```
*-                                     Procedure Exit                                -*
```

```
*-                                                                              -*
```

```
*- Purpose : Do all that is necessary to Exit the current form.                -*
```

```
*- Called From : Valid Event of Exit Button.                                  -*
```

```
*-                                                                              -*
```

```
*- Written By: Dr. Thomas E. Hicks                                           App: Visual FoxPro 5 -*
```

```
*-----*
```

```
Procedure Exit
```

```
* Release all screens in this application
```

```
    __RestForm.release
```

```
* Two Lines Below Required For Compiled Applications
```

```
*Clear Read
```

```
*quit
```

```
Cancel
```

```
Return
```

```
EndProc
```

8] The Code to associate with procedure browse is found below:

```

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*-
*-                                     Procedure Browse                                     -*
*-                                     -*
*- Purpose : Establish a browse selection for this application.                        -*
*-                                     -*
*- Written By: Dr. Thomas E. Hicks                                     App: Visual FoxPro 5 -*
*-_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*
Procedure Browse
    Browse NoAppend NoDelete NoModify ;
        Title "Move Cursor To Desired Record - Hit The ESC Key"
    _screen.activeform.refresh
Return
EndProc

```

9] The Code to associate with button Delete is found below:

```

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*-
*-                                     Procedure UnDelete                                     -*
*-                                     -*
*- Purpose : Launch Undelete form from which user can select the                      -*
*-           a record from which the deletion tag can be removed.                    -*
*-           Procedure AcceptUnDelete or Procedure CancelUndelete                    -*
*-           will be called from the Undelete form.                                  -*
*- Called From : Valid Event of UnDelete Button.                                    -*
*-                                     -*
*- Written By: Dr. Thomas E. Hicks                                     App: Visual FoxPro 5 -*
*-_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*
Procedure UnDelete
    Set Deleted Off                                     && Show Deleted Records
    __LastFilter = filter()                             && Save Environment Info
    __CurrentLoc = RecNo()
    Set Filter To Deleted()                             && Select only deleted records
    Count to NoRecords                                  && Count will contain # deleted records

    If NoRecords = 0 Then                               && If there are no deleted records
        Wait Window "No Records Available to Undelete" + chr(7) Timeout 10
        Set Filter To &__Lastfilter
        Set Deleted On
        Goto __CurrentLoc
    Else                                                 && If there are deleted records
        __RestForm.hide
        Do Form Forms/Undelete
    Endif
Return
EndProc

```

10] The Code to associate with button Cancel on the Undelete form is found below:

```

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*-
*-                                     Procedure CancelUnDelete                                     -*
*-                                     -*
*- Purpose : Terminate the UnDelete form and return to status quo.                  -*
*- Called From : The Cancel button of form UnDelete.                              -*
*-                                     -*
*- Written By: Dr. Thomas E. Hicks                                     App: Visual FoxPro 5 -*
*-_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*
Procedure CancelUnDelete
    Set Deleted On
    Set Filter To &__LastFilter
    GoTo __CurrentLoc
    _screen.activeform.release
    __RestForm.show
Return
EndProc

```

11] The Code to associate with button Undelete on the Undelete form is found below:

```
*-*****-
*-                               Procedure AcceptUnDelete                                -*
*-                                                                                      -*
*- Purpose : Terminate the UnDelete form, undelete the selected                    -*
*-           record and return to the previous status. Keep the                     -*
*-           undeleted form in view.                                                -*
*- Called From : The UnDelete Now button of form UnDelete.                        -*
*-                                                                                      -*
*- Written By: Dr. Thomas E. Hicks                                           App: Visual FoxPro 5 -*
*-*****-
Procedure AcceptUnDelete
    Recall
    Set Deleted On
    Set Filter To &__LastFilter
    _screen.activeform.release
    __RestForm.show
Return
EndProc
```

12] The Code to associate with button Search is found below:

```

*-----*
*                                     Procedure Search                                     *
*-----*
* Purpose : Launch Search form from which user can select the                       *
*           a) what key to search (SearchCombo)                                   *
*           b) what population to search (SelectCombo)                             *
*           c) what to search for (SearchText)                                     *
*           SearchExample shall display an example search string.                  *
*           Procedure SetSearchExample helps to match the search                    *
*           example with the selection. It is called from the                      *
*           the valid event of the SearchCombo.                                   *
*           Procedure AcceptSearch is called from the Search Now                   *
*           button of the Search form.                                             *
*           Procedure CancelSearch is called from the Cancel                      *
*           button of the Search form.                                             *
* Called From : Valid Event of Search Button.                                     *
*-----*
* Written By: Dr. Thomas E. Hicks                                           App: Visual FoxPro 5 *
*-----*
Procedure Search
    __Example = " "
    __SearchNo = 1
    __RestForm.hide
    Do Form Forms/Search

Return
EndProc

```

13] The Code to associate with button cancel on the Search form is found below:

```
*-----*
*-                               Procedure CancelSearch                               -*
*-                                                                           -*
*- Purpose : Cancel the Search form.                                           -*
*- Called From : Valid Event of Cancel Button on form Search.                 -*
*-                                                                           -*
*- Written By: Dr. Thomas E. Hicks                                           App: Visual FoxPro 5 -*
*-----*
Procedure CancelSearch
    _screen.activeform.release
    __RestForm.show
    _screen.activeform.refresh
Return
EndProc
```

14] The Code to associate with button Search Now on the Search form is found below:

```

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*-                               Procedure AcceptSearch                               -*
*-                               -*
*- Purpose : Search for the __SoughtValue. This generic search                    -*
*-           form will have to convert text data to numeric, date,                -*
*-           or logical when appropriate.                                         -*
*-           This will be a near match search. The previous                      -*
*-           environment will be restored when finished.                         -*
*- Called From : Valid Event of Search Now Button on form Search.                -*
*-                               -*
*- Written By: Dr. Thomas E. Hicks                                           App: Visual FoxPro 5 -*
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
Procedure AcceptSearch
    If __SoughtValue = " " Then
        Wait Window "There Is No Search Data - Try Again" Timeout 10
    Else
        __LastOrder = Order()
        __LastFilter = Filter()
        __CurrentLoc = RecNo()

        Do Case
            Case __SearchNo = 1
                Set Order To Name
                __SoughtValue = AllTrim(Upper(__SoughtValue))
            Case __SearchNo = 2
                Set Order To Phone
                __SoughtValue = AllTrim(__SoughtValue)
        EndCase
        Seek __SoughtValue
        If EOF( ) Then
            GoTo Bottom
        EndIf
        Set Filter To &__LastFilter
        Set Order To &__LastOrder
        __screen.activeform.release
        __RestForm.show

Return
EndProc

```

15] The Code to associate with button SetSearchExample on the Search form is found below:


```
*-----*
*-                               Procedure SetSearchExample                               -*
*-                                                                                         -*
*- Purpose : shall display an example search string. It helps to                      -*
*-           match the search example with the selection.                            -*
*- Called From : Valid Event of the SearchCombo.                                     -*
*-                                                                                         -*
*- Written By: Dr. Thomas E. Hicks                                           App: Visual FoxPro 5 -*
*-----*
Procedure SetSearchExample
    Do Case
        Case __SearchNo = 1
            __Example = "Search For : Long Horn Cafe"
        Case __SearchNo = 2
            __Example = "Search For : 111-2222"
    EndCase
    _screen.activeform.refresh

Return
EndProc
```

16] The Code to associate with the Select Combo is found below: In order to more carefully identify my reports, I add a report subtitle to each report for clarification. This is most easily done when the filter is set! I save the last filter and last filter number for restoration purposes. It enables me to restore the environment in situations where I might have just run a report for Texas restaurants.

```
*-*****-*  
*-                               Procedure ProcessFilterNo                                -*  
*- Purpose : Process the Filter Selection options for this                            -*  
*-           application.                                                                -*  
*-                                                                                      -*  
*- Written By: Dr. Thomas E. Hicks                                           App: Visual FoxPro 5 -*  
*-*****-*  
  
Procedure ProcessFilterNo  
  
    __CurLocation = RecNo()  
Do Case  
    Case __FilterNo = 1                && Everybody  
        Set Filter To  
            __ReportTitle = " "  
            __ReportSubTitle = " "  
  
    Case __FilterNo = 2                && State = Texas  
        Set Filter To Upper(State) = "TX"  
        __ReportSubTitle = " [Texas]"  
  
    Case __FilterNo = 3                && State = Florida  
        Set Filter To Upper(State) = "FL"  
        __ReportSubTitle = " [Florida]"  
  
    Case __FilterNo = 4                && State = Oklahoma  
        Set Filter To Upper(State) = "OK"  
        __ReportSubTitle = " [Oklahoma]"  
  
    Case __FilterNo = 5                && State = West Virginia  
        Set Filter To Upper(State) = "WV"  
        __ReportSubTitle = " [West Virginia]"  
EndCase
```

Continuation of SetFilterNo. Note that we should not allow the user to set a filter that provides zero records.

```

Count To NoRecords
If (NoRecords = 0) Then          && If No matches for this filter
    Set Filter to &__LastFilter  && tell user and restore to previous
    __FilterNo = __LastFilterNo
    GoTo __CurLocation          && filter condition
    ? chr(7)
    wait Window "There Are No Matches For This Record At This Time " Timeout 15
    _screen.activeform.refresh
Else                             && Make sure that a record of this filter
    Do Next                     && is currently in view. There is a better
    Do Previous                 && way to do this!
    __LastFilter = Filter()
    __LastFilterNo = __FilterNo
EndIf
Return
EndProc

```

17] The Code to associate with the Order Combo is found below: I save the last order and last order number for restoration purposes. It enables me to restore the environment in situations where I might have just run a report for Texas restaurants in order by name.

```

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*-                               Procedure ProcessOrderNo                               -*
*- Purpose : Process the Index/Order options for this                               -*
*-                               application.                                         -*
*-                               -*                                                  -*
*- Written By: Dr. Thomas E. Hicks                               App: Visual FoxPro 5 -*
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
Procedure ProcessOrderNo
    Do Case
        Case __OrderNo = 1          && Index By Name
            Set Order To Name

        Case __OrderNo = 2          && Index By Phone
            Set Order To Phone

    EndCase
    __LastOrderNo = __OrderNo
    __LastOrder = Order( )
    _screen.activeform.refresh

Return
EndProc

```

18] The Code to associate with button Print is found below:

```

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*-*
*-*                               Procedure Print                               -*
*-*
*-* Purpose : Launch Print form from which user can select the             -*
*-*           a) what form or label to print (PrintCombo)                  -*
*-*           b) what population to print (All, 1, Rest Radio)              -*
*-*           Procedure AcceptPrint is called from the Print Now            -*
*-*           button of the Print form.                                     -*
*-*           Procedure CancelPrint is called from the Cancel               -*
*-*           button of the Print form.                                     -*
*-* Called From : Valid Event of Print Button.                             -*
*-*
*-* Written By: Dr. Thomas E. Hicks                                         App: Visual FoxPro 5 -*
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
Procedure Print
    __PrintNo = 1
    __PrintOptionNo = 1
    __RestForm.hide
    Do Form Forms/Print
Return
EndProc

```

19] The Code to associate with the Cancel button on the Print form is found below:

```

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*-*
*-*                               Procedure CancelPrint                       -*
*-*
*-* Purpose : Close Down the Print form and return to the main form        -*
*-* Called From : Valid Event of Cancel Button on Print Form Button        -*
*-*
*-* Written By: Dr. Thomas E. Hicks                                         App: Visual FoxPro 5 -*
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
Procedure CancelPrint
    __screen.activeform.release
    __RestForm.show
Return
EndProc

```

20] The Code to associate with the Print button on the Print form is found below: Unfortunately there is an error in the distribution of the Visual Studio 6 that prevents the rest of the record option from working.

```

*-----*
*-                               Procedure AcceptPrint                               -*
*-                               -*
*- Purpose : Build and Execute an Appropriate Print String.                        -*
*- Called From : Valid Event of Cancel Button on Print Form Button                -*
*-                               -*
*- Written By: Dr. Thomas E. Hicks                               App: Visual FoxPro 5 -*
*-----*
Procedure AcceptPrint
    __PrintOptionString = " "
    __CurLocation = RecNo()
    Do Case
        Case __PrintNo = 1          && Print Full Report
            __PrintOptionString = "Report Form Reports\Complete "
            __ReportTitle = "Complete Report " + __ReportSubTitle

        Case __PrintNo = 2          && Print Name Phone Report
            __PrintOptionString = "Report Form Reports\NamePhone "
            __ReportTitle = "Name Phone Listing " + __ReportSubTitle

        Case __PrintNo = 3          && Print Mailing Labels
            __PrintOptionString = "Label Form Reports\Mail "
            Set Order To Zip
    EndCase

    Do Case
        Case __PrintOptionNo = 1    && All Records
            __PrintOptionString = __PrintOptionString + "To Print NoConsole"

        Case __PrintOptionNo = 2    && This Record Only
            __PrintOptionString = __PrintOptionString + " Next 1 To Print NoConsole"

        Case __PrintOptionNo = 3    && Rest Of The Records
            __PrintOptionString = __PrintOptionString + " Rest To Print NoConsole"
    EndCase

```

Continuation Of AcceptPrint

```

__screen.activeform.release
& __PrintOptionString
GoTo __CurLocation
Set Order To & __LastOrder
__screen.activeform.refresh
__RestForm.show

Return
EndProc

```

Add Tool Tips To All Your Buttons!

1] In addition to the graphic, each of your buttons should have extremely clear tips. (See Below!)

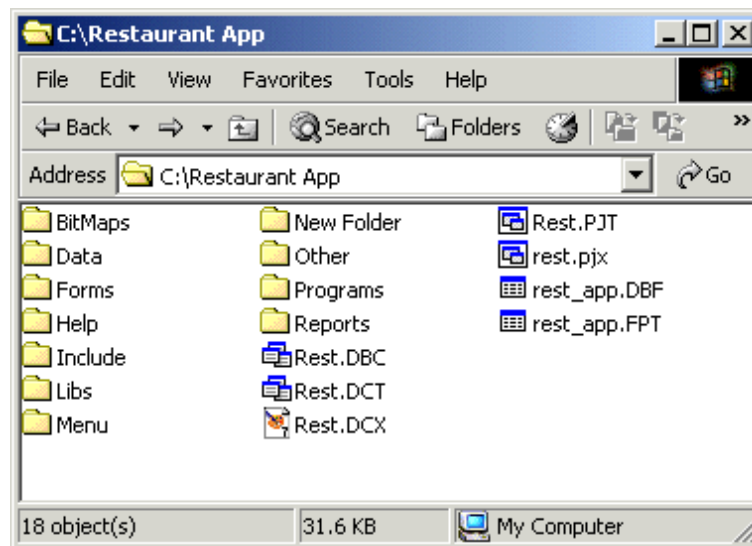
Messages for these buttons shall be:

- Go To The First Record
- Go To The Last Record
- Go To The Next Record
- Go To The Previous Record
- Add New Record
- Delete Record
- Undelete Record
- Browse
- Save
- Cancel
- Search
- Edit
- Print
- Exit

Build An Executable

With an executable, the user will not have to launch Foxpro or know any Foxpro coding/procedures/etc.

1] Your RestarurantApp folder probably looks something like the following:



2] The code in the previous Exit module did not really exit foxpro. It allowed you to get an idea of what your application was going to look like. It is now time to correct the Exit module

```

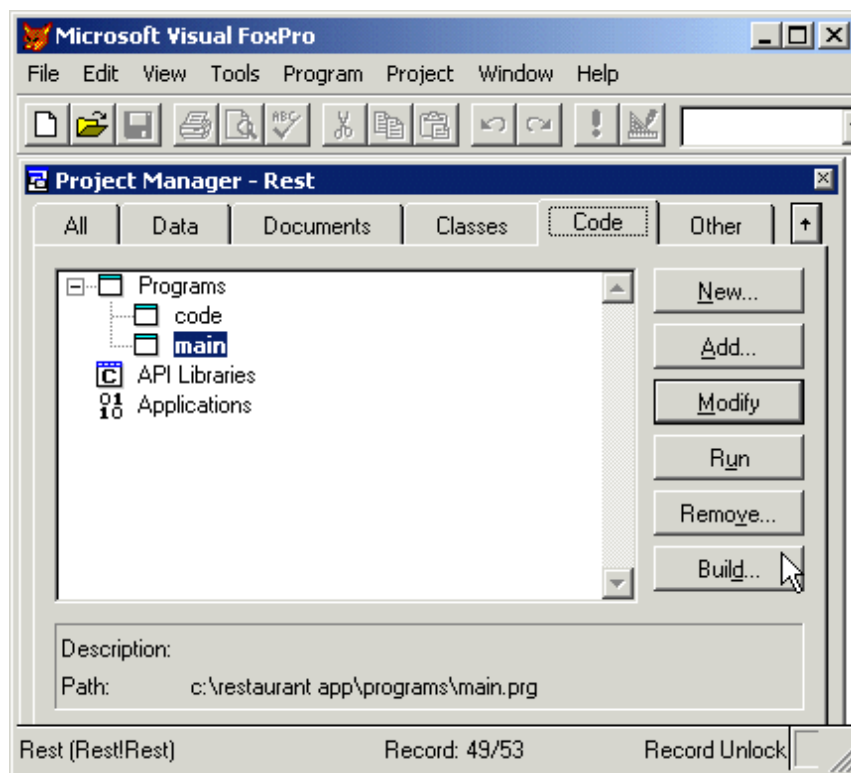
*-----*
*-                               Procedure Exit                               -*
*-                               -*
*- Purpose : Do all that is necessary to Exit the current form.             -*
*- Called From : Valid Event of Exit Button.                               -*
*-                               -*
*- Written By: Dr. Thomas E. Hicks                               App: Visual FoxPro 5 -*
*-----*
Procedure Exit
* Release all screens in this application
  __RestForm.release

* Two Lines Below Required For Compiled Applications
Clear Read
quit

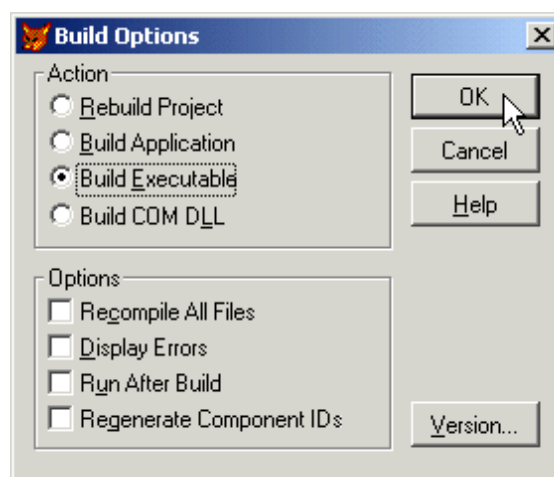
*Cancel
Return
EndProc

```

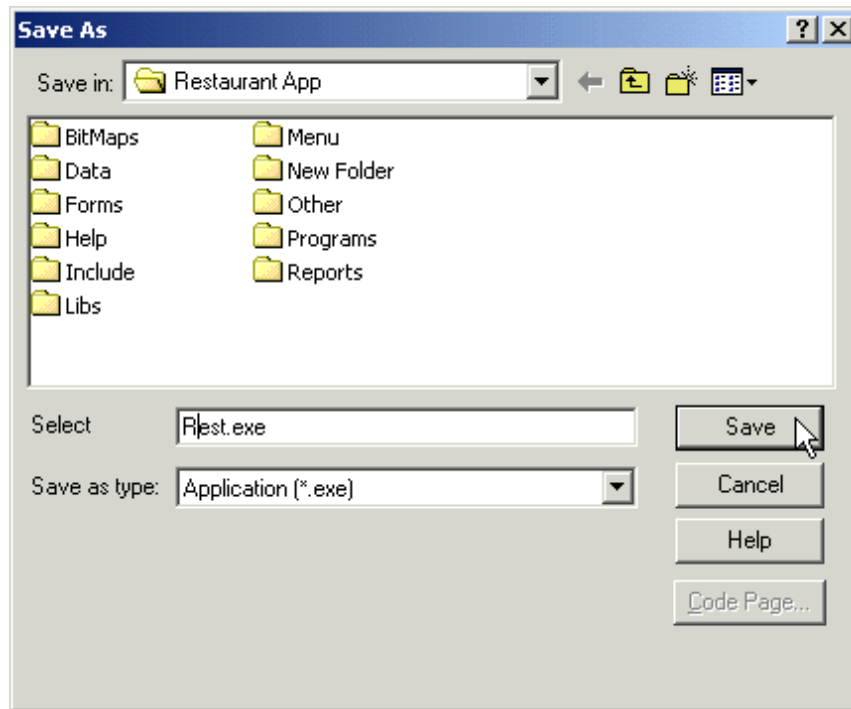
3] Using the mouse, select the Code tab of your project. Using the mouse, select the main program. Using the mouse select/push the **Build** button. (See Below!)



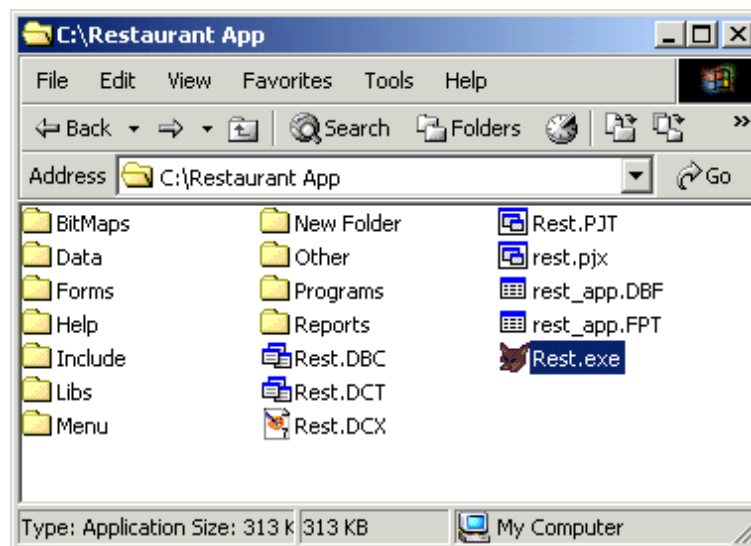
4] Using the mouse, select the Build Executable radio button. Using the mouse select/push the **OK** button. (See Below!)



5] Name the executable Rest.exe. Using the mouse select/push the **Save** button. (See Below!)

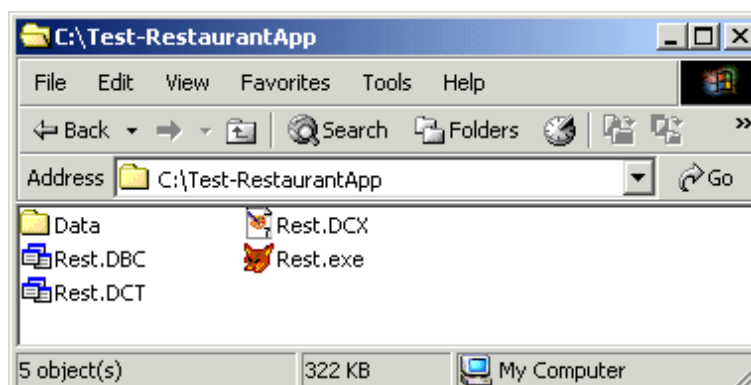


6] The executable is now complete. You may execute it by double-clicking on it. Try it! (See Below!)



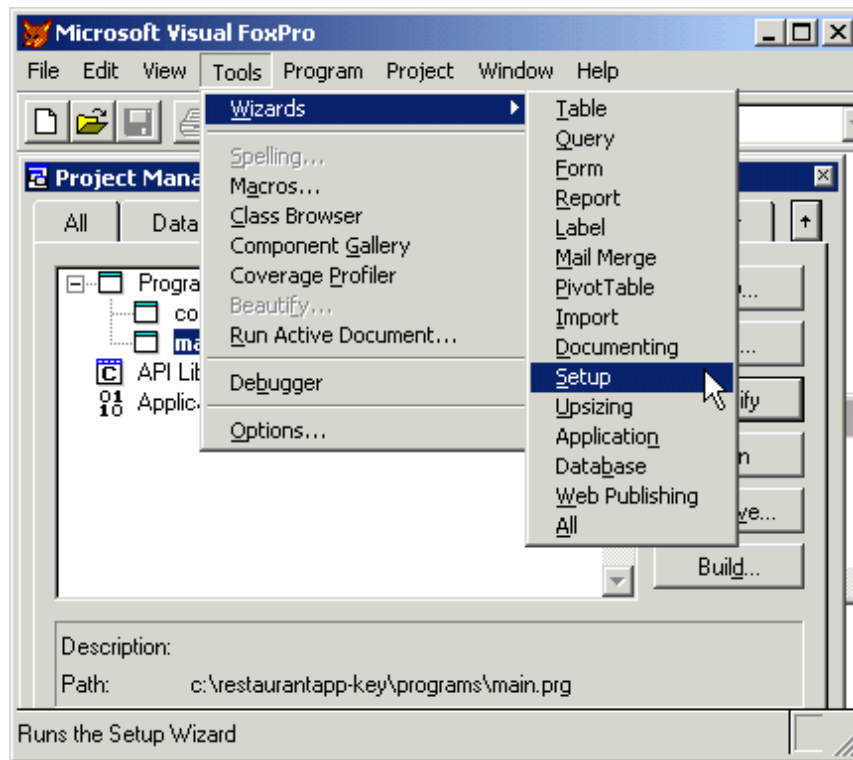
What Files Do I Really Need For Program Execution?

1] The files necessary for execution are the application, the database, and the data. Since it is best to install all applications in the Program Files directory, I would normally recommend placing this folder in the Program Files Directory before creating the install disks; you will see why shortly! I would also recommend renaming the folder title appropriately ==> such as Restaurants.

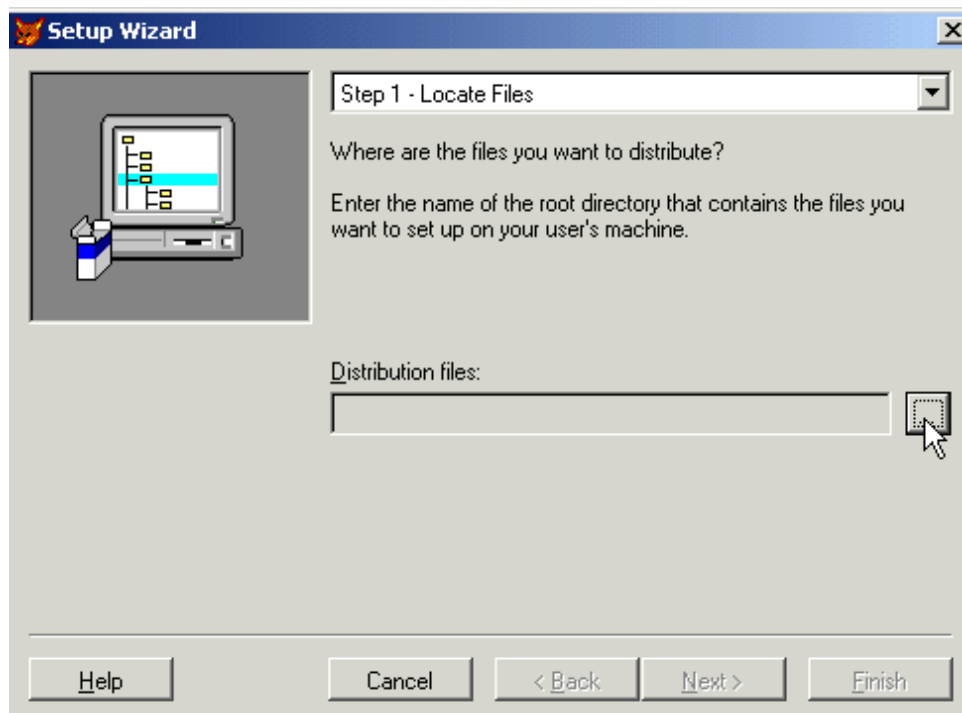


Preparing Installation Disks!

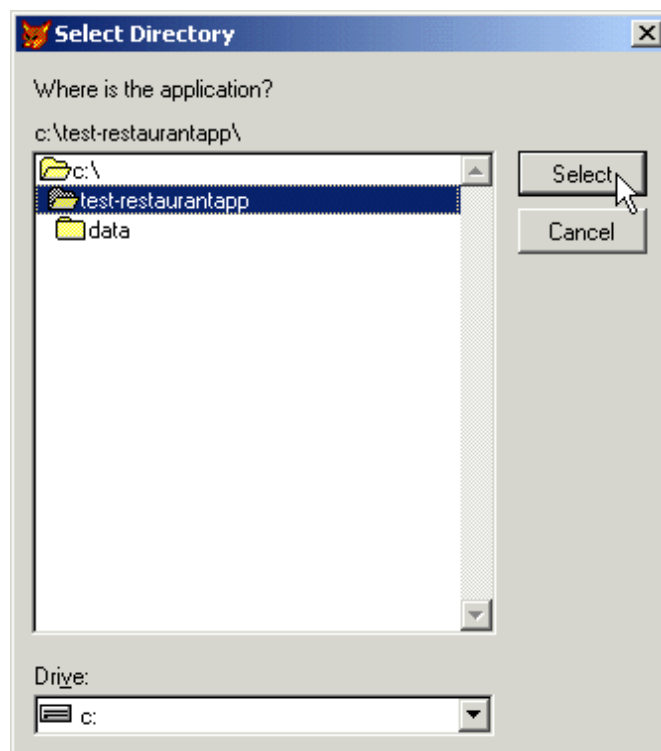
1] Using the mouse, hold down the Tools menu bar ==> Select Wizards ==> Select Setup. (See Below!)



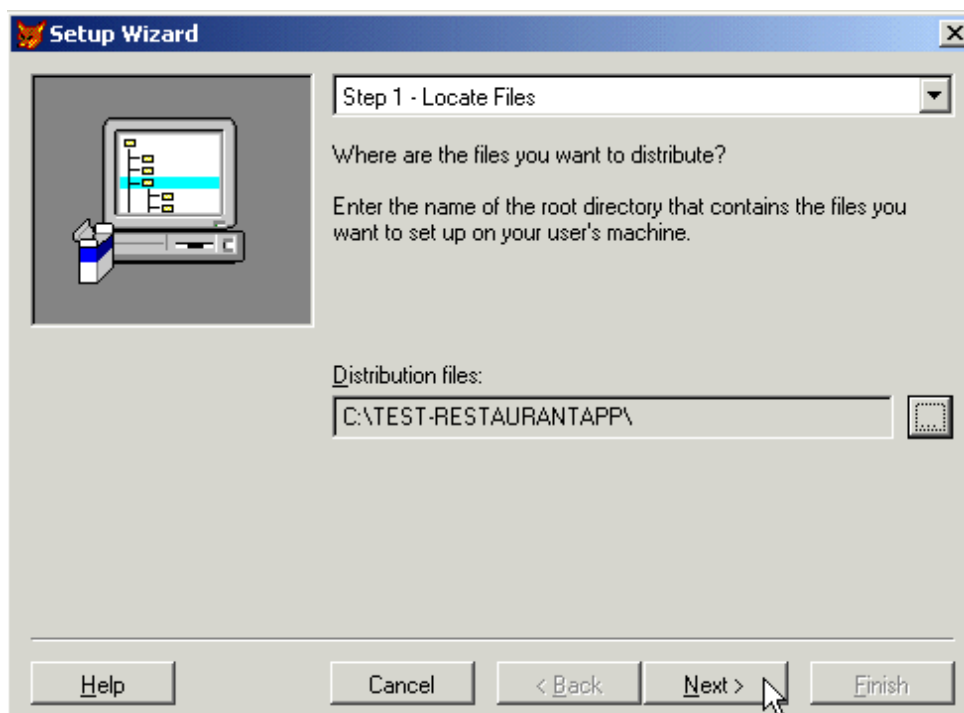
2] Push the file access button to select the distribution files. Once more, I recommend placing them in the Program Files directory. (See Below!)



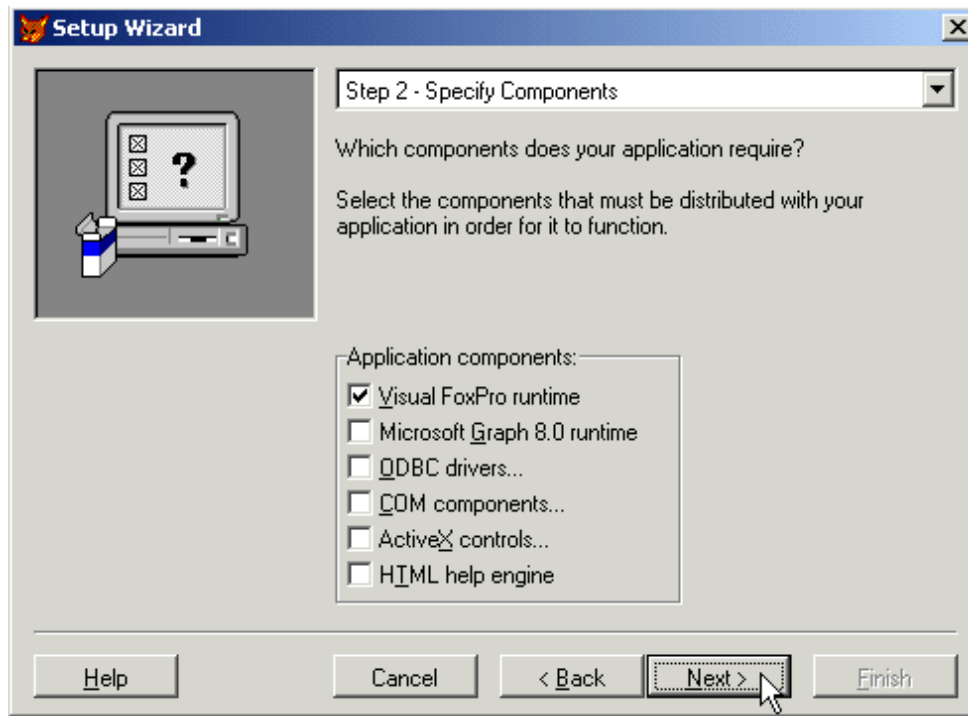
3] As you perhaps noticed on the screen capture above, I have placed my distribution files in C:\Test-RestaurantApp folder. Navigate to the proper folder, and select/push the **Select** button. (See Below!)



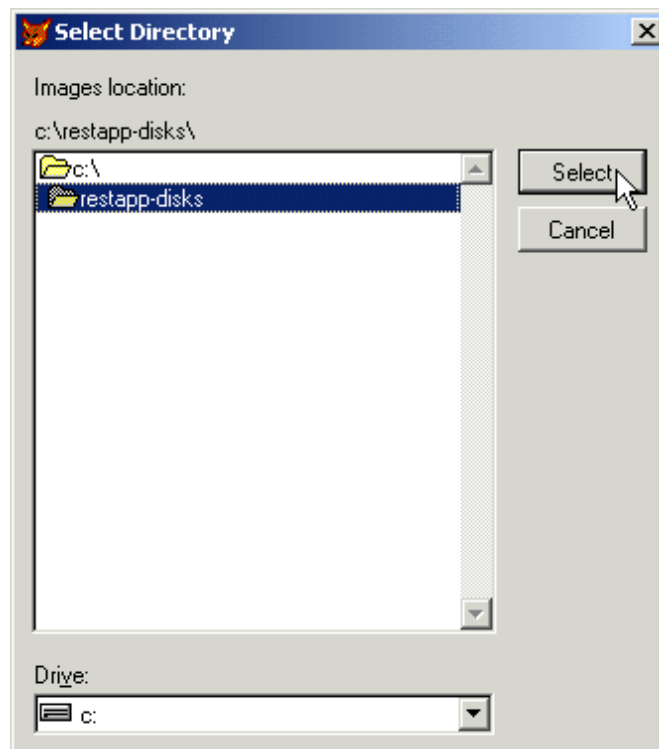
4) Using the mouse, select/push the **Next** button. (See Below!)



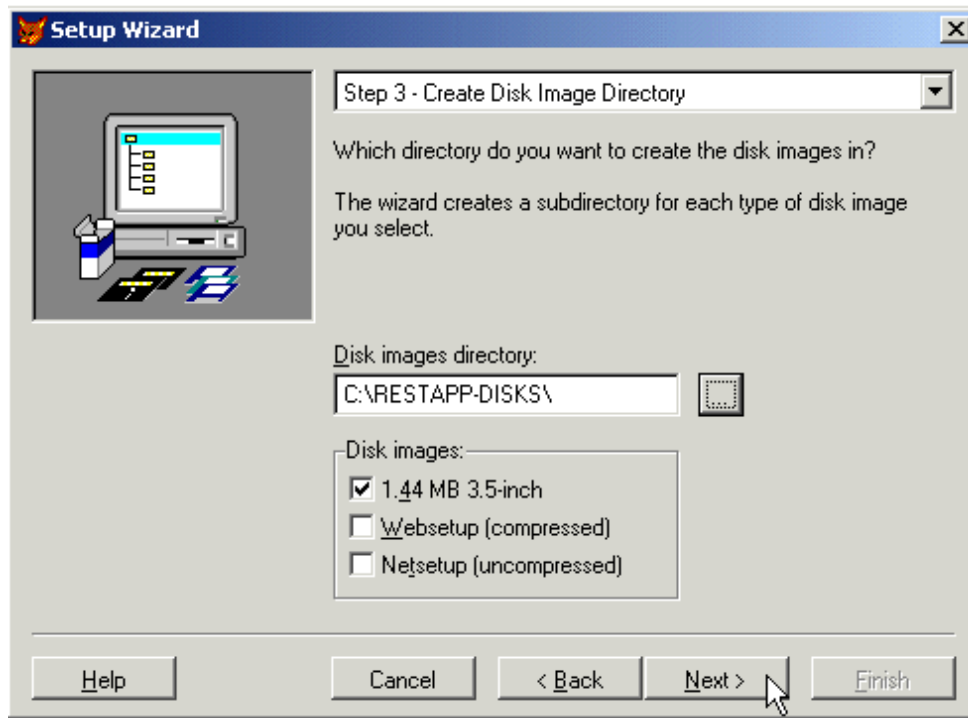
5) Using the mouse, check Visual FoxPro runtime components. Using the mouse, select/push the **Next** button. (See Below!)



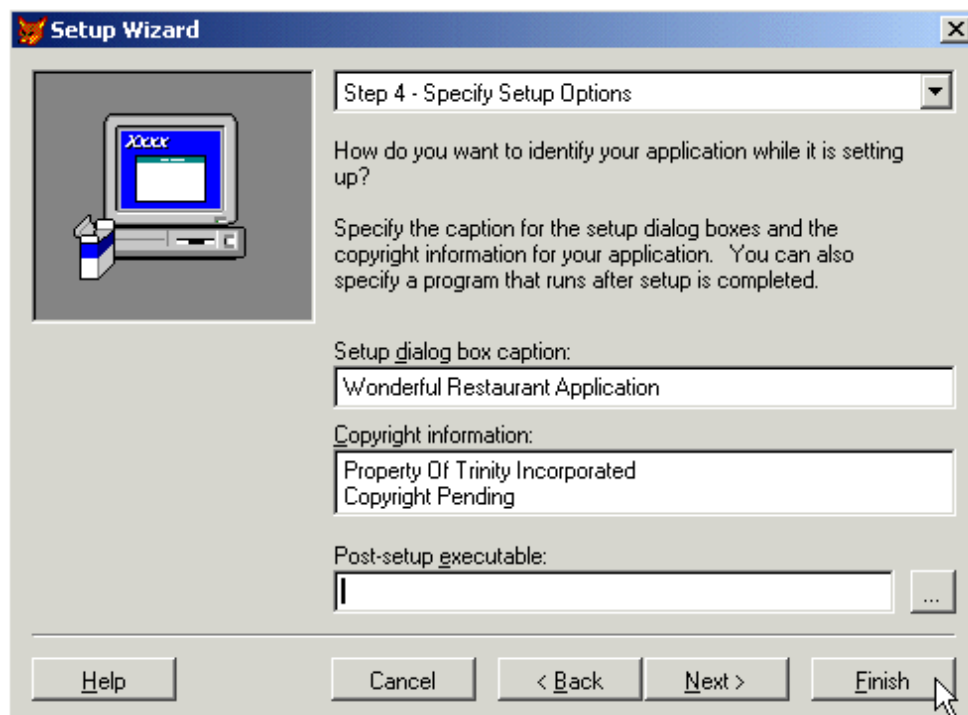
6] You may place your distribution disk in any existing folder. I have prepared, in advance, a folder C:\RestApp-Disks for this purpose. Navigate to the desired folder. Using the mouse, select/push the **Select** button. (See Below!)



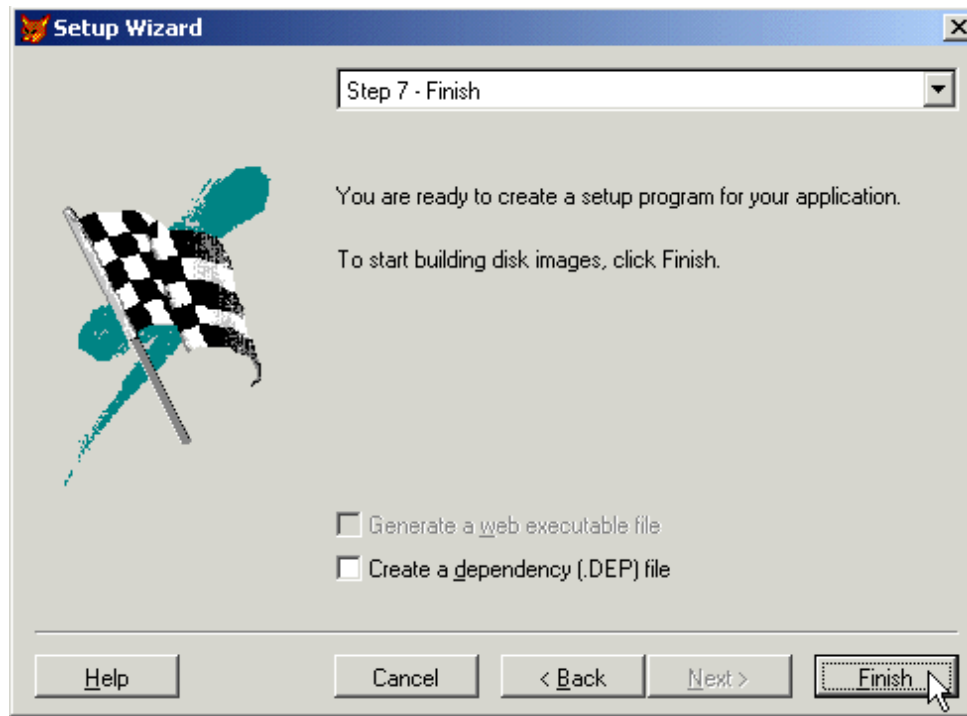
7] Using the mouse, select/push the **Next** button. (See Below!)



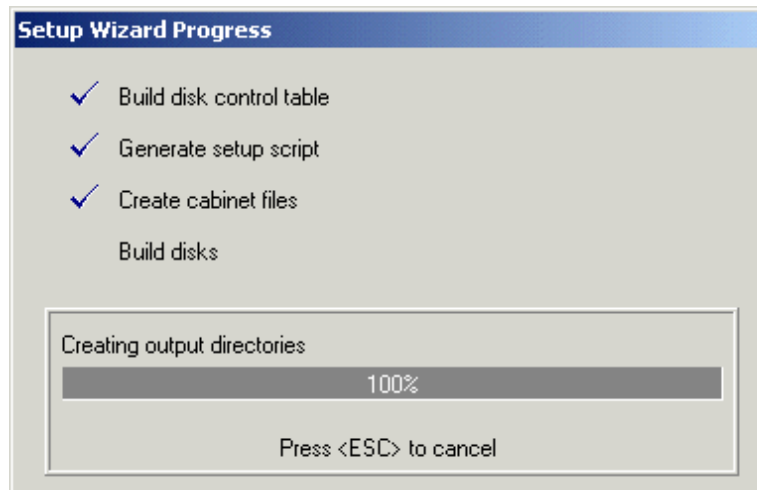
8] The Setup dialog box caption is vital; select something appropriate. Notice how many times you see this string during the install process. Copyright is optional as is appropriate. You can select a program to execute upon completion. Using the mouse, select/push the **Next** button. (See Below!)



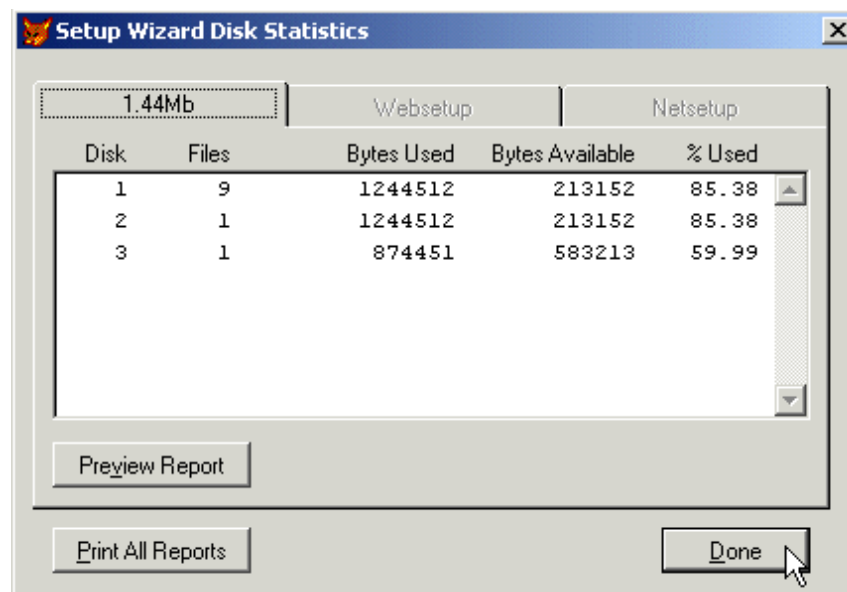
9] Using the mouse, select/push the **Finish** button. (See Below!)



10] It starts to work! Does not take long.

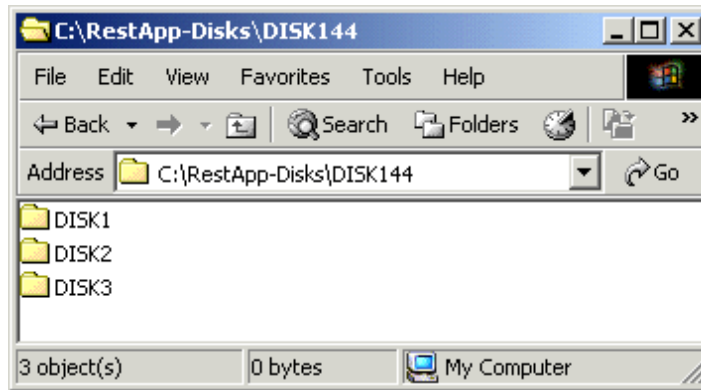


11] It will require 3 floppy disks to store those run-time DLL's necessary to run this foxpro application on a system that has neither FoxPro nor another FoxPro application.



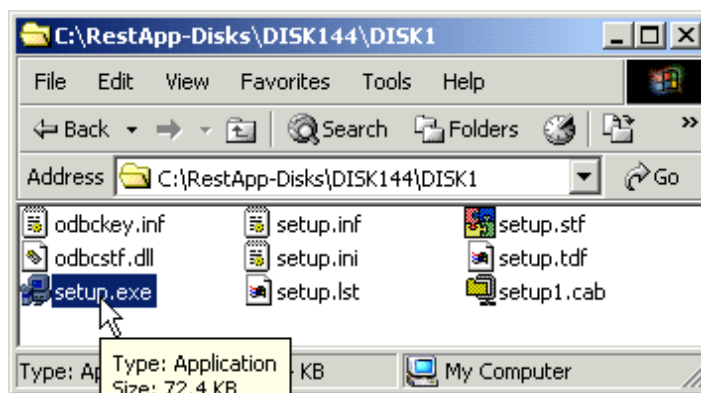
12] Created for you in the desired location is a folder called DISK144. You may copy DISK1 to the root of floppy disk #1 and DISK2 to the root of floppy disk #2 and DISK3 to the root of floppy disk #3; this will enable you to install from floppy

disks.

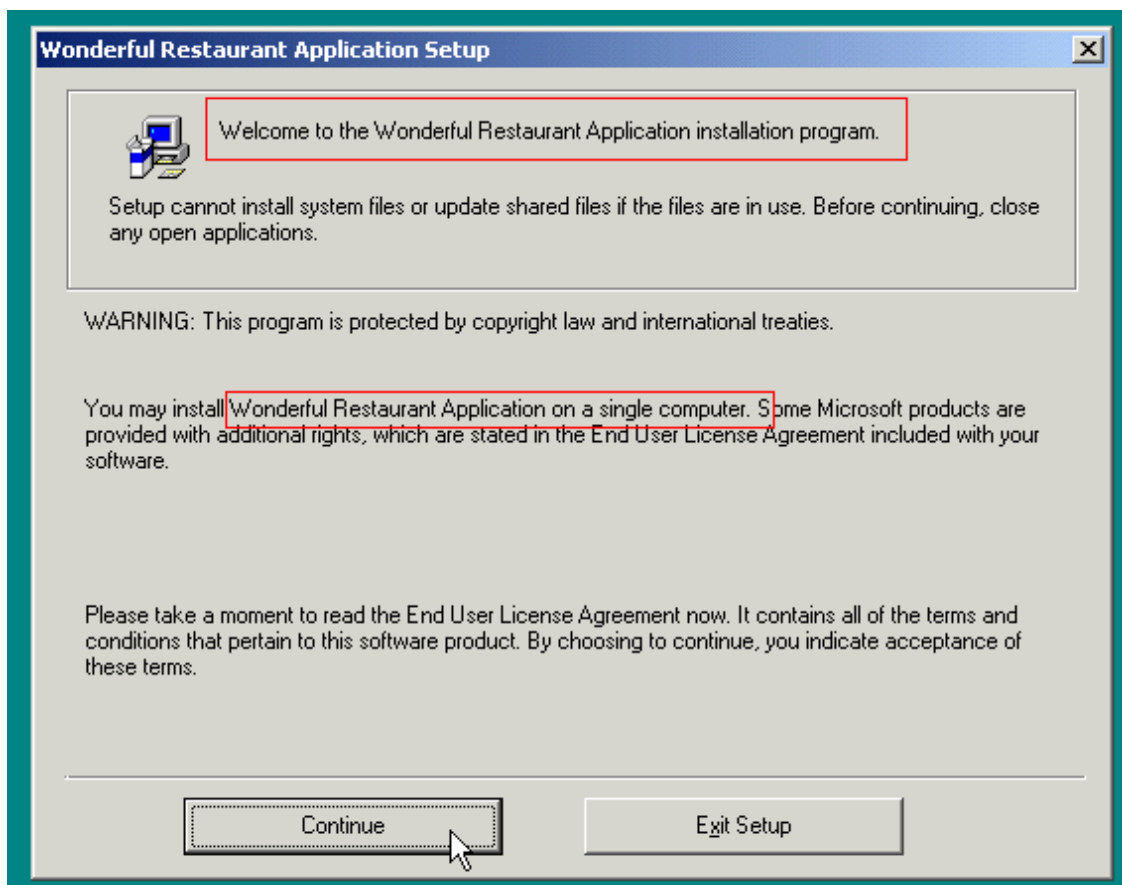


Install The Application From Install Disks Or Network

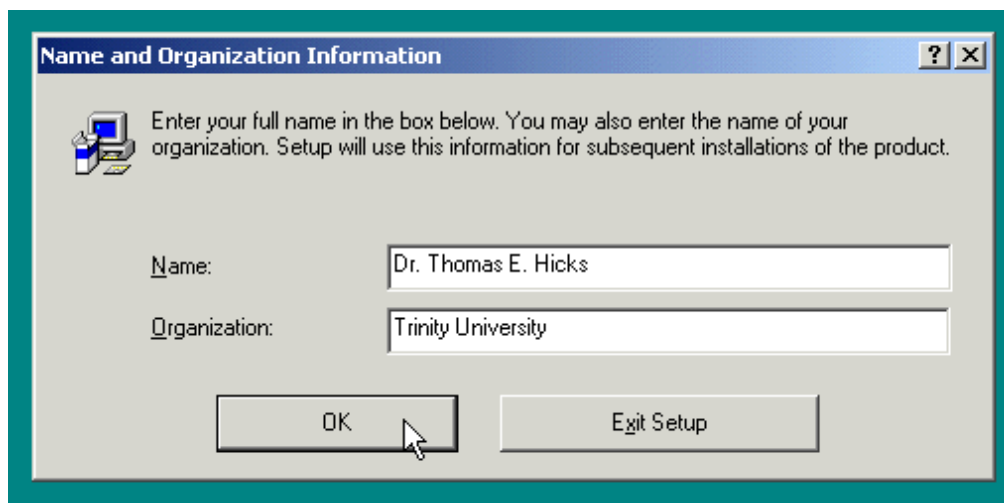
1] You may also simply leave these three folders intact and install from the network. In either event, double-clicking on Setup.exe will install the application. (See Below!)



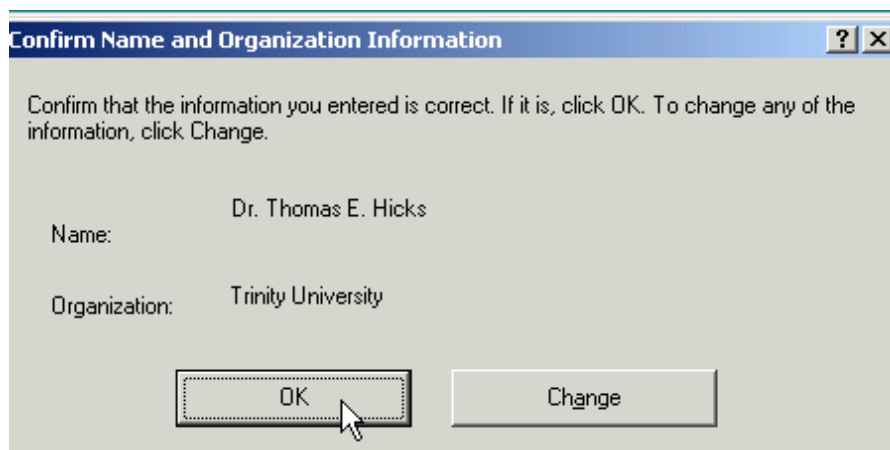
2] Note the Setup Dialog Caption. Using the mouse select/push the **Continue** button.



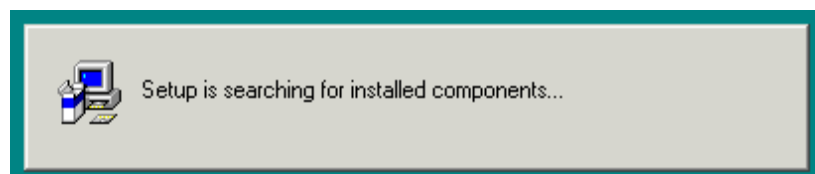
3] It pulls info from previous windows install.



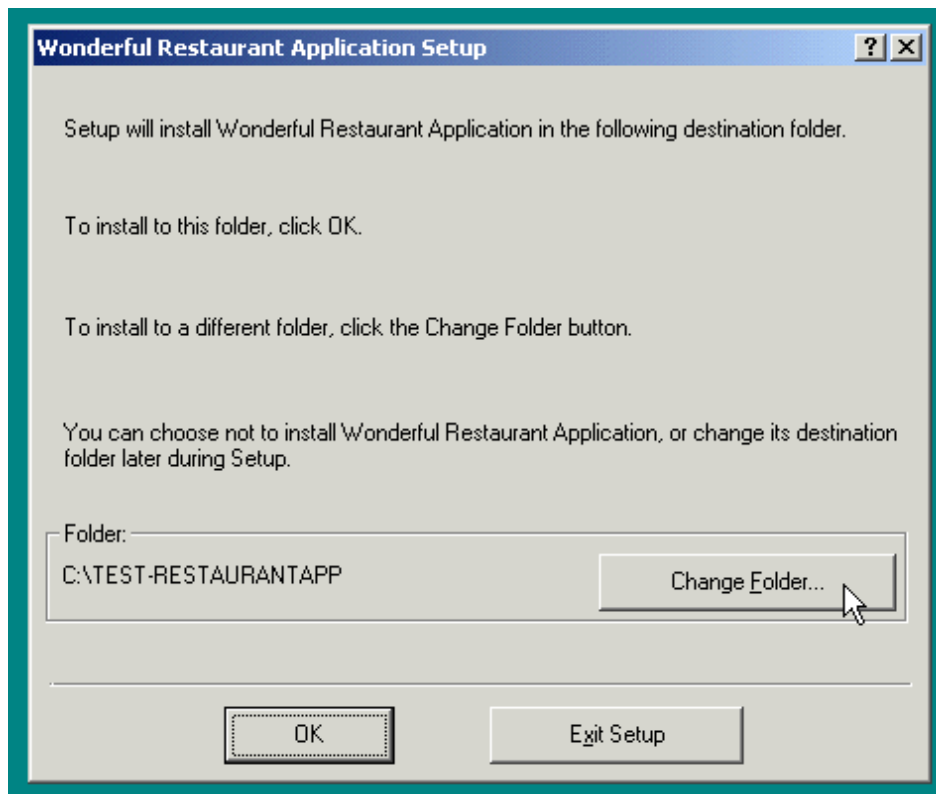
4] Using the mouse select/push the **OK** button.



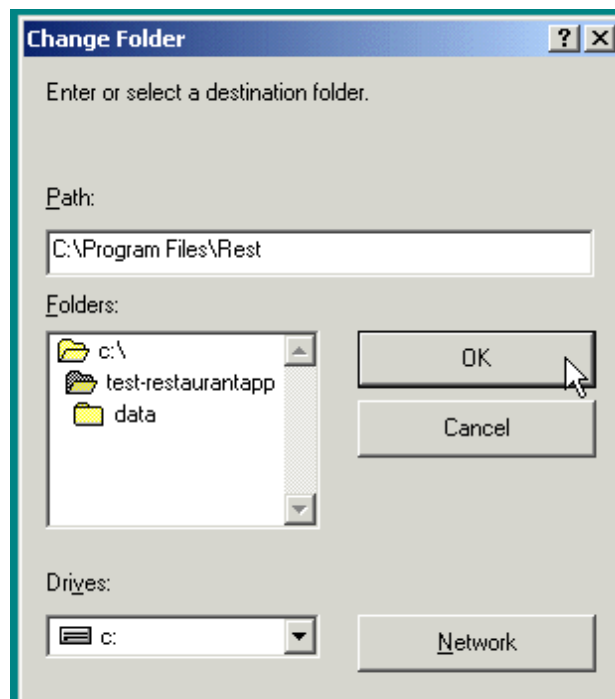
5] Working!



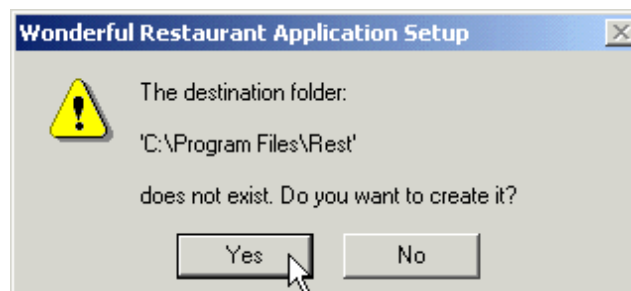
6] It would have been so much less trouble for the user if I had created the installation disks from the Program Files directory. A knowledgeable user will want to change the location. Using the mouse select/push the **Change Folder** button.



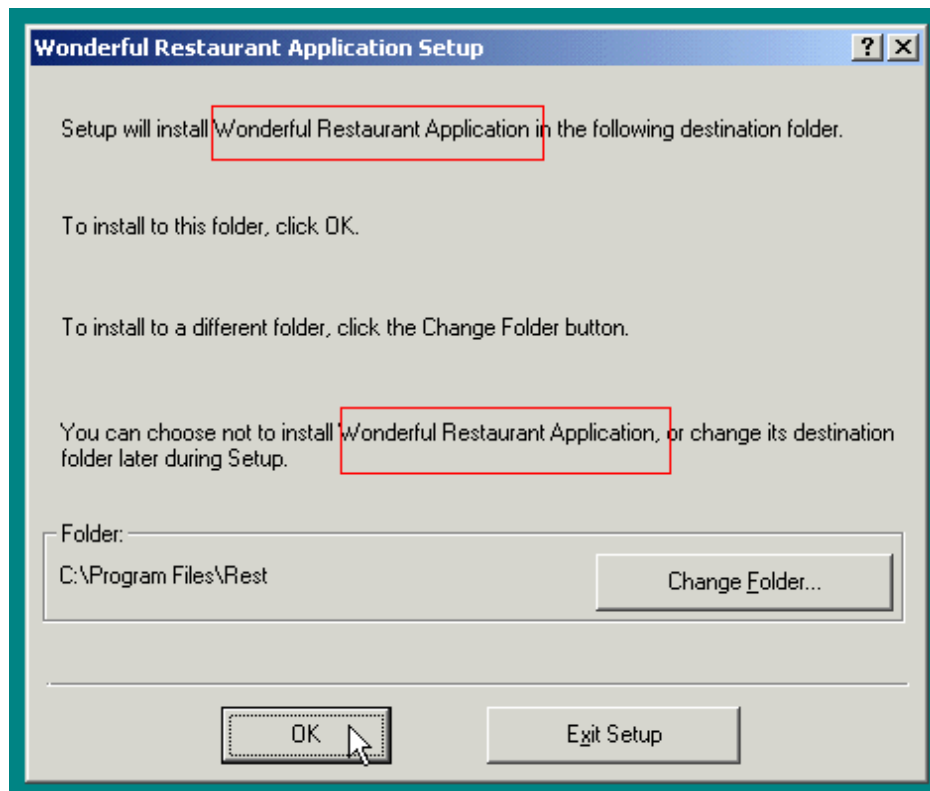
7] Select Program Files directory and give the folder a decent title. Using the mouse select/push the **OK** button.



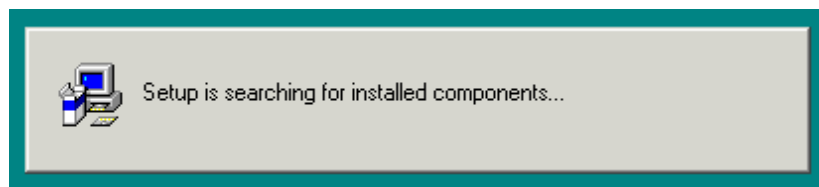
8] Using the mouse select/push the **Yes** button.



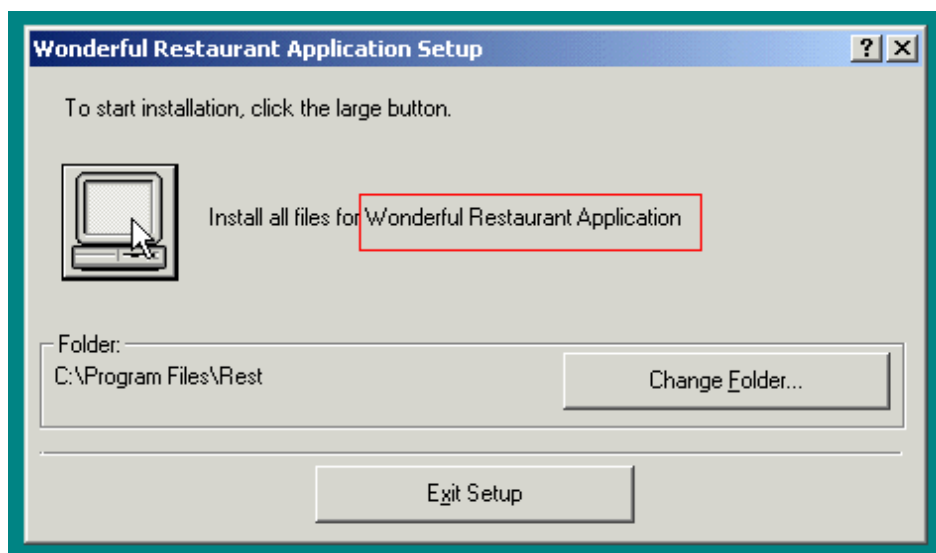
9] Using the mouse select/push the **OK** button.



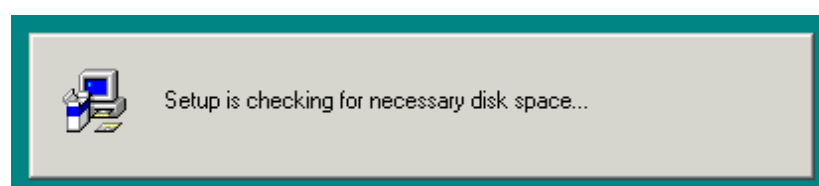
10] Working!



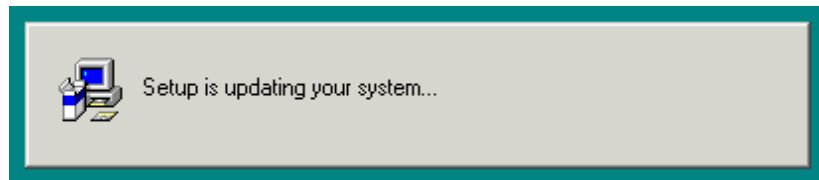
11] Using the mouse select/push the **Application Installation** button.



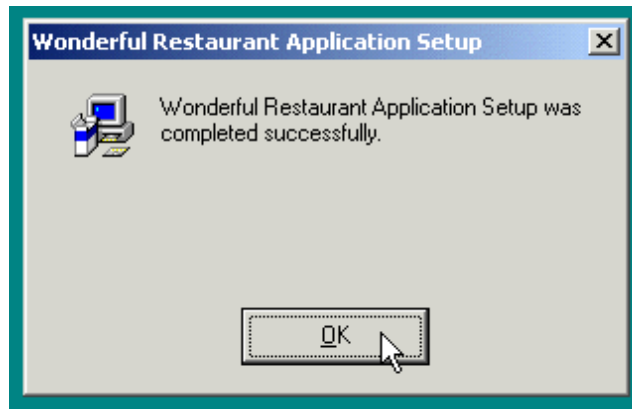
12] Working!



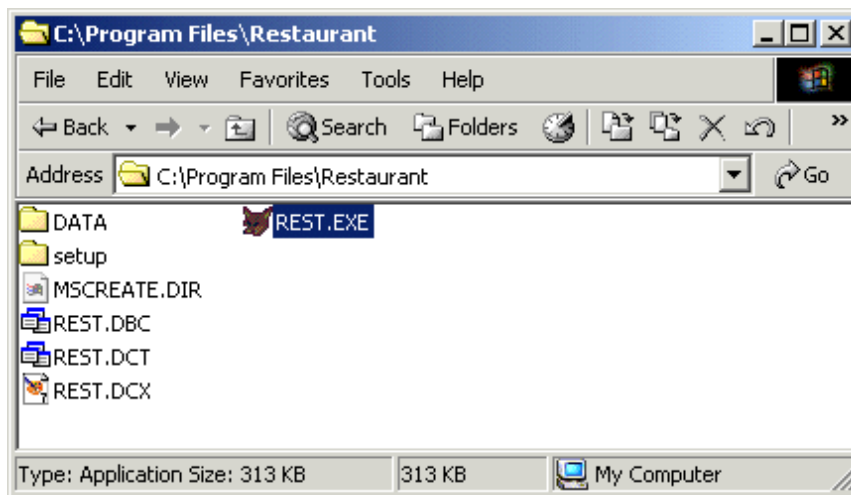
13] Still Working!



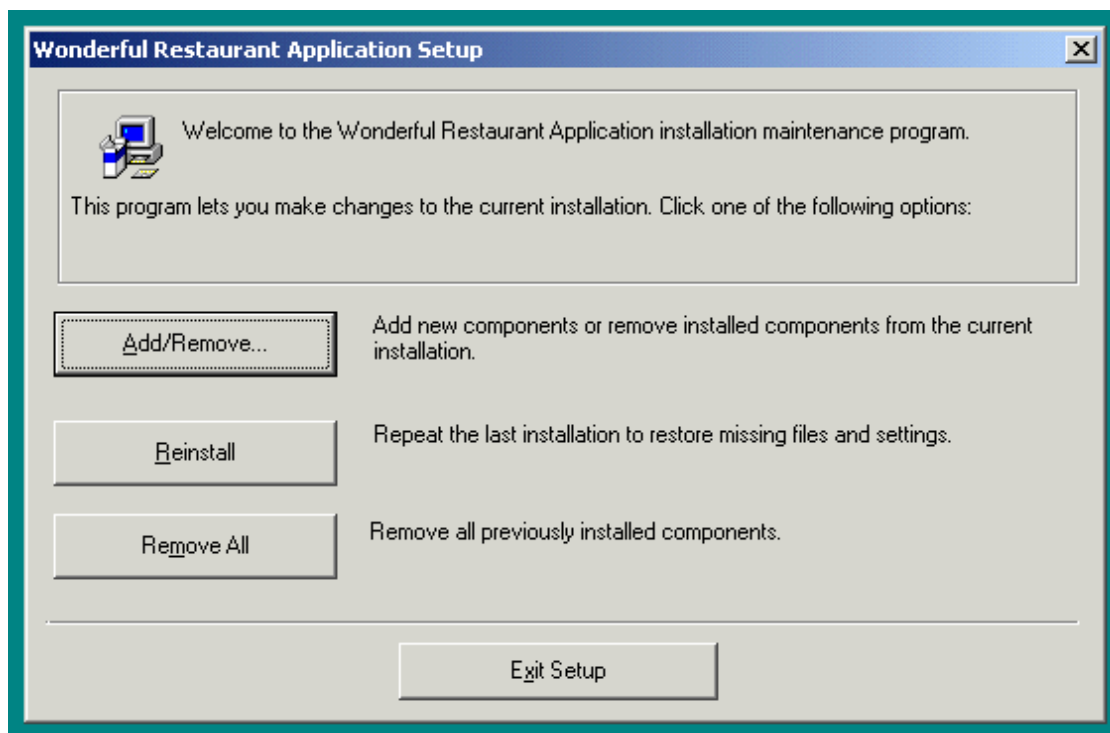
14] Using the mouse select/push the **OK** button.



15] The application now exists in folder [C:\Program Files\Restaurant](#). You may launch it by double-clicking on it. You may create a short cut to it and place it in the start menu or on the desktop. Try it!



Go Back And Re-Launch the Setup To Un-Install or Re-Install



May be accessed through URL: <http://www.cs.trinity.edu/~thicks>

May also be accessed through URL: <http://carne.cs.trinity.edu>

This Document May Not Be Printed or Reproduced Without Written Permission.

2003 Copyright : Dr. Thomas E. Hicks

Permission granted : Professional Educators & College Students may print one copy of this page!



Dr. Thomas E. Hicks

Computer Science Department

Trinity University

"Dr. Web"

Dr. Web's Site

Computer Science

Send Mail

Trinity University